THE ROLE OF CANALIZATION IN THE SPREADING OF PERTURBATIONS IN BOOLEAN NETWORKS

Santosh Venkatiah Sudharshan Manicka

Submitted to the faculty of the University Graduate School
in partial fulfillment of the requirements
for the degree
Doctor of Philosophy
in the School of Informatics and Computing,
Indiana University
May, 2017

Accepted by the Graduate Faculty, Indiana University, in partial fulfillment of the requirements for the degree of Doctor of Philosophy.

Doctoral Committee

Luis M. Rocha, PhD

Randall D. Beer, PhD

Olaf Sporns, PhD

Yong-Yeol Ahn, PhD

April 24, 2017

Acknowledgments

I would like to thank my advisor Prof. Luis Rocha for making this entire dissertation happen, and for teaching me the scientific method of research and the best practices in academic writing; I am indebted to him for that. I would like to thank committee members Profs. Randall Beer, Yong-Yeol Ahn and Olaf Sporns for investing valuable time and effort in reading and critiquing this dissertation, and for helping me see novel ways of extending my research. I would like to extend my hearty thanks to Prof. Martin Siegel for seeing me through a predicament that could have pushed me off-course. A special thanks to beloved Linda Hostetter for supporting my progress throughout by promptly informing me about course registrations, missed grades and many other important administrative matters. My hearty thanks to my former colleagues Dr. Artemy Kolchinsky and Dr. Alexander Gates for insightful discussions during various stages of my research, and also for helpful comments on a draft of a paper I wrote on the third chapter. I would like to thank CASCI and CNetS members for providing feedback on my presentations of my research. I would also like to thank the FLAD Computational Biology Collaboratorium at the Gulbenkian Institute (Portugal) for supporting my research during my summer stints at the institute in 2011 and 2012, and the IGERT program (Indiana) for supporting almost all of my summer research. Finally, to say that I thank my parents, Dr. Sudharshan Venkatiah and Radha Sudharshan, and close friends, Dr. Aarti Veernala and Alex Jose, for their support during difficult times would be an understatement; no words can express my feelings of gratitude toward them.

Santosh Venkatiah Sudharshan Manicka

THE ROLE OF CANALIZATION IN THE SPREADING OF PERTURBATIONS IN BOOLEAN NETWORKS

Canalization is a property of Boolean automata that characterizes the extent to which subsets of inputs determine (canalize) the output. Here, we investigate the role of canalization as a characteristic of perturbation-spreading in random Boolean networks (BN) with homogeneous connectivity via numerical simulations. We consider two different measures of canalization introduced by Marques-Pita and Rocha, namely 'effective connectivity' and 'input symmetry', in a three-pronged approach. First, we show that the mean 'effective connectivity', a measure of the true mean in-degree of a BN, is a better predictor of the dynamical regime (order or chaos) of the BN than the mean in-degree. Next, we combine effective connectivity and input symmetry in a single measure of 'unified canalization' by using a common yardstick of Boolean hypercube "dimension", a form of fractal dimension. We show that the unified measure is a better predictor of dynamical regime than effective connectivity alone for BNs with large in-degrees. When considered separately, the relative contributions of the two components of the unified measure changes systematically with the mean in-degree, where input symmetry becomes increasingly more dominant with larger in-degrees. As an application, we show that the said measures of canalization characterize the dynamical regimes of a suite of Systems biology models better than the in-degree. Finally, we introduce 'integrated effective connectivity' as an extension of effective connectivity that characterizes the canalization present in BNs with arbitrary timescales obtained by iteratively composing a BN with itself. We show that the integrated measure is a better predictor of long-term dynamical regime than just effective connectivity for a small class of BNs known as the elementary cellular automata. This dissertation will advance theoretical understanding of BNs, allowing us to more accurately predict their short-term and long-term dynamic character, based on canalization. As BNs are generic models of complex systems, combining interaction graphs with multivariate dynamics, these results contribute to the complex networks and systems field. Moreover, as BNs are important models of choice in Systems biology, our methods contribute to the burgeoning toolkit of the field.

Luis M. Rocha, PhD

Randall D. Beer, PhD

Olaf Sporns, PhD

Yong-Yeol Ahn, PhD

Contents

Curriculum Vitae

Chapter 1

Canalization in the Dynamics of Complex Systems

1.1 Overview

Networks are popular models of choice for many natural and technological systems. Patterns in network topology have yielded deep insights into the systems being modeled. For example, the apparent universality of scale-free topology of networked natural systems (e.g., cellular regulation, brain networks) has lent support to the idea that different systems may have similar governing mechanisms underlying their architecture and behavior [11–13]. Multivariate dynamical systems can be seen as complex networks that come with an additional specification about how their nodes (variables) behave dynamically. They are more general because they, unlike graphs, accept more than pairwise interactions. The *transfer functions* that specify node behavior can be non-linear functions linking inputs and output in complex (e.g., non-additive) ways. As a result, the role of overt topology in the overall dynamic behavior of the multivariate system may only be partial, thus warranting systematic considerations of the role of transfer functions.

Boolean networks (BN) are exemplars of complex discrete dynamical systems that were originally introduced as models of genetic regulatory systems [65, 67]. The Boolean ON-OFF idealization captures the characteristic switch-like behavior of gene expression [48, 49, 67].The Boolean logic functions model the *combinatorial regulation*, also known as *synergy*, observed in many natural biochemical systems including cell signaling and metabolic networks [3,46,129]. The Boolean modeling approach, also known as 'logic modeling', has been successful to the extent that it has become possible to synthetically alter the 'regulation logic' of genes in real gene regulatory systems [22], in such a way as to dramatically alter the gene expression patterns [104]. Thus, logic modeling has proven to be useful both as an approach to understanding the overall organization of a biochemical system and making predictions about its dynamics and controllability, even while certain important physical properties like biochemical rates, concentration and timing are not considered [1,3,17,23,139,140]. It has morever produced models able to faithfully reproduce the full trajectory of states from a known initial condition to the attractor, e.g., the yeast cell cycle model [17].

BN have also found applications in a variety of other fields including neuroscience, social science and epidemiology [3, 9, 34, 93, 101]. In these cases, Boolean networks serve as modes of spreading and communication processes. That is, they help model how infections, ideas, voting patterns and neuronal signals diffuse. The question typically posed about these models is: how do perturbations spread in the long run[1]?

One of the most active areas of research in complexity dealing with multivariate dynamical systems is the topic of *criticality*, in which case we are typically interested in characterizing how small perturbations spread in the short-term, in other words, whether they amplify or dissipate. Analyses of a number of Boolean network (BN) models of gene regulation have shown that they are either dynamically stable or operate near the critical boundary separating the ordered and the chaotic regimes [10, 48, 49, 118]. In the ordered regime, the system is stable in response to most perturbations, whereas in the chaotic regime it is sensitive to perturbations. Thus, the intermediate *critical* regime is expected to achieve an optimal balance between robustness and flexibility [5]. Many stud-

---

[1] the definition of the term 'perturbation' depends on the study (e.g., an infection in epidemics models is equivalent to a perturbation)

ies concur with the idea that natural biochemical systems exhibit criticality, that is, they exhibit dynamics associated with the critical regime [99, 110, 127]. A few biochemical systems, however, exhibit dynamics more typical of the ordered regime [118]. For example, the segment polarity network of *Drosophila* is known to be highy robust to perturbations [88, 136]. Our own analysis of a BN repository known as 'Cell collective' reveals a few examples of biochemical systems that reside deeper in the ordered regime (Fig. 2.13 in Ch.2). It is nevertheless thought that criticality confers many functional advantages to the system, for example, better defense in bird flocks, maximal amplification of a wide range of frequencies in the auditory system, ability to explore of a wide range of states in neural systems and optimization of information flow during the morphogenesis of the Drosophila embryo [71, 99] and in cortical networks [16].

The criticality of a system can be measured empirically — in either statistical physics terms (long-range correlations, non-Gaussian and power-law distributions of fluctuations etc.) or in dynamical systems terms (neighboring trajectories tend to neither converge nor diverge with time). A *theory* of criticality allows us to predict whether a system is critical or not based on some of its characteristic features (e.g., modularity, average degree distribution, transfer function properties etc.). Theories of criticality of RBNs are useful because they prescribe a set of *sufficient* conditions for a BN to occupy a certain dynamical regime. Consequently, they lead to a better understanding of the possible influence of those constraints on the dynamics of complex systems and how they have evolved. A theory of criticality could also be useful in a clinical setting—assuming accurate systems biology models of biochemical regulation [130]—where it may help constrain the set of possible interventions for, say, driving a system from a 'diseased' state to a 'healthy' state. Naturally, 'better' theories prescribe more precise and reliable constraints for interventions. A good case in point is cancer. Cancer can be thought of as the appearence of a stable pattern of genetic expression—a 'stable misbehavior'—in the epigenetic landscape of a cell [57, 105]. A recent study further suggests that a potentially universal characteristic of cancer is hyper-variability in the genetic expression profiles [20]. These observations have led to the hypothesis that cancerous systems may be slightly chaotic in nature since presence of many, if stable, attractors is a telltale sign of chaos in Boolean networks [106]. Thus, interventional prescriptions for triggering apoptosis (normal cell death) in cancerous cells based on Boolean network models of such systems (e.g., [140]) may be better guided by more accurate theories of criticality.

One of the earliest theories of criticality for Boolean networks, introduced by Stuart Kauffman, was also one of the simplest: networks with more than two inputs on the average tend to be chaotic, otherwise are critical or stable [67]. However, it is not the most accurate, since it is known that a number of gene regulatory systems have many more than an average of just two inputs, yet are not chaotic [118]. A more accurate theory, put forth by Derrida [32], considered both the number of inputs and the 'output bias' of a node (the probability with which a node outputs a 1 or 0, or is ON or OFF, for a randomly chosen input) as its tuning parameters. While allowing networks with a relatively large number of inputs to be stable, the theory still required that the average output bias be very high, which is uncharacteristic of real systems as well [4]. Aldana addressed this issue by introducing a theory of criticality for scale-free Boolean networks that accommodated both very large connectivity and low average output bias in the stable regime [4].

A multitude of theories of criticality is in principle possible for BN since they can be characterized in different ways (average in-degree, output bias and scale-free exponent etc.) [67]. It is thus natural to ask what features characterize actual biochemical systems and motivate theories with the same. A striking feature of gene regulatory systems is that they tend to be regulated by highly 'canalizing' transfer functions [10, 48, 49, 67, 126, 127]. *Canalization* is a property of a Boolean function wherein the state of a single *canalizing* input is sufficient to determine the output [67]. If a gene has a canalizing transcription factor, for example, then it alone can determine whether the gene should be transcribed or not even while other factors are present or not. A Boolean function with one or more such canalizing inputs is known as a *canalizing function.* It has been shown that the fraction of nodes with canalizing functions is a more reliable indicator of the dynamical regime of the network than its mean in-degree and output bias [62]. As a consequence, a larger fraction of canalizing transfer functions tends to push a network into the stable regime [10, 64]. A more restricted form of canalization, known as 'nested canalization', describes Boolean functions with hierarchical layers of canalization, which have been found to be a characteristic of certain biochemical systems [63]. Moreover, the number of canalization layers, quantified as the "nested canalizing depth", is also known to be an indicator of the dynamical regime of a BN [60,75]. Another measure of canalization, known as *sensitivity*, takes a probabilistic form: it is the expected number of inputs of a Boolean function whose individual toggling results in the toggling of the output [117]. The mean sensitivity of a random BN is also known to determine its dynamic regime [113].

The types of canalization described above measure the extent to which *single* inputs determine the output of Boolean functions. In this sense, they can be viewed as measures of *linear* canalization since the expected degree of canalization of a *set* of inputs would at best be, all else being equal, a linear combination of the degrees of canalization of single inputs. Marques-Pita and Rocha [88] have recently introduced a more general, nonlinear, measure of canalization called *input redundancy*—as well as the complementary *effective connectivity.* These are measures of *partial canalization* where subsets of inputs are allowed to be *jointly* canalizing. Thus, for instance, a pair of inputs may be jointly canalizing even if the individual inputs are not (e.g., the XOR function). Partial canalization is known to be prevalent in Boolean functions [108]. Marques-Pita and Rocha have also introduced a higher-order form of partial canalization known as *input symmetry* that quantifies the extent to which the states of subsets of inputs equally determine (canalize) the output [87, 88]. The inputs to the logical OR function, for example, are symmetric, since as long as one of the inputs is ON the output is ON. The relationship between partial canalization and criticality has not been explored before, and is a focus of this dissertation.

Criticality is a characterization of short-term spreading of perturbations, as mentioned before. It is thus natural to ask if the aforementioned measures of canalization are predictive of perturbation-spreading over longer time intervals as well. Given the nonlinear nature of a typical BN, it is reasonable to suppose that they may not be appreciably predictive. The reason is that it is typically not possible to predict the long-run behavior of nonlinear dynamical systems using analytical methods [124], meaning that the characteristic "static" features (e.g., the degree of canalization) are typically not predictive as well since analytical solutions necessarily comprise such features. It is, however, possible to compute the long-term behavior of a nonlinear dynamical system through numerical integration [124]. This motivates the question of whether a Boolean analog of "integration" where a BN is iteratively composed with itself may lead to the formulation of an even more nuanced measure of canalization—the *integrated effective connectivity*—as a predictor of the long-term dynamical behavior of BNs.

The goal of this dissertation is to characterize the relationship between canalization and criticality, and perturbation spreading in general, in Boolean networks. We focus on the following three questions:

1. Is a theory of criticality based on mean effective connectivity (or input redundancy) a better predictor of the dynamic regime of a BN than one based on mean in-degree, as pursued by current theories?
2. Does the prediction of dynamical regime improve by incorporating both input redundancy and symmetry into a unified measure of canalization?
3. Is a measure of 'integrated effective connectivity', indicating the mean effective connectivity of an "integrated" BN obtained by iteratively composing a BN with itself, a better predictor of perturbation-spreading over long intervals of time than the mean effective connectivity of the original BN?

The basic motivation behind these questions is that the aforementioned measures of partial canalization of Boolean functions are more nuanced than the conventional measures, and therefore may be more predictive of the dynamical response to perturbations, resulting in more accurate theories of criticality and of perturbation spreading in general.

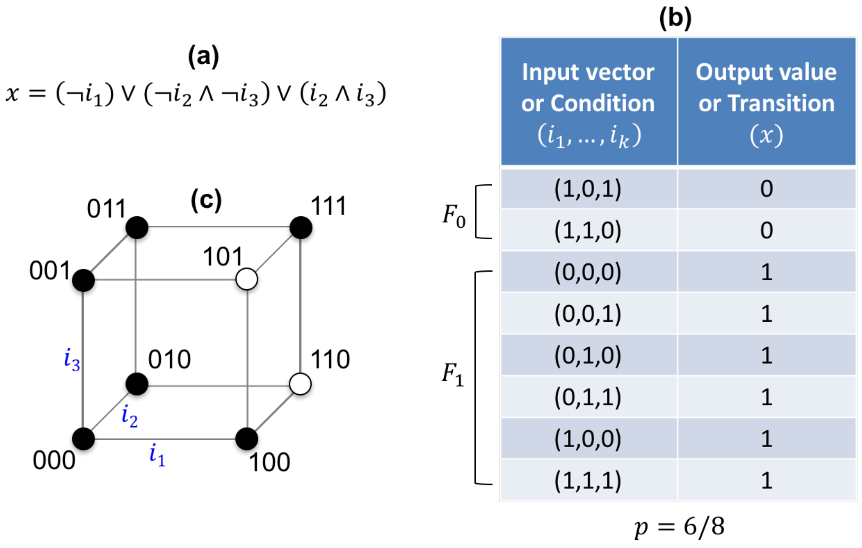
In the following sections, the conceptual building blocks that support this dissertation, and their theoretical foundations are described in detail.

1.2 Boolean automata

A *Boolean automaton* is a binary variable, $x \in \{0, 1\}$, where state 0 is interpreted as *false* (*off* or *unexpressed*), and state 1 as *true* (*on* or *expressed*). The states of x are updated in discrete time-steps, t, according to a *Boolean state-transition function* or simply *Boolean function* $f : \{0,1\}^k \mapsto \{0,1\}$ of k inputs variables: $x = f(i_1, \ldots, i_k)$. Such a function can be defined by a *Boolean logic formula* or by a *look-up (truth) table* (LUT) with 2^k entries each containing an input vector and the corresponding output value. An example of the former is $x = f(i_1, i_2, i_3) = (\neg i_1) \vee (\neg i_2 \wedge \neg i_3) \vee (i_2 \wedge i_3)$; $\wedge$, $\vee$ and $\neg$ denote logical conjunction, disjunction, and negation respectively. Each variable or its negation in the formula is known as a *literal*, and a term within a set of parentheses is known as a *conjunctive clause* as the literals are joined by logical conjunction. The entire formula is a *disjunctive clause* since the terms are joined by logical disjunction. This standard representation of logic formulas is known as the *disjunctive normal form* (DNF) [28]; all logic formulas in this dissertation are described in DNF form. The states of x are updated in discrete time steps, so $x^{t+1} = \neg(i_1^t) \vee (\neg i_2^t \wedge \neg i_3^t) \vee (i_2^t \wedge i_3^t)$, in the above example.The LUT for this example function is shown in Fig. 1.1b. Each LUT entry, f_α, is defined by a *condition* (input vector), denoted by a Boolean vector of length k, and the *transition* (output value), denoted by a Boolean scalar. The entire LUT is denoted as: $F \equiv \{f_\alpha : \alpha = 1, \ldots, 2^k\}$. The subsets of the

LUT that map to the outputs 0 and 1 are denoted as: F_0 and F_1, respectively, so $F \equiv F_0 \cup F_1$; $F_0 \cap F_1 = \phi$.

A graphical representation of this function is depicted as a *Boolean hypercube* in (Fig. 1.1(c)), where the Euclidean coordinates of the corners represent the input vectors f_α (LUT entries), and the color of the circle at each corner represents the output (black = 1, white = 0). The *output bias* of the function, p, is the proportion of outputs equal to 1: $p = \frac{|F_1|}{2^k}$.



(a)

$x = (\neg i_1) \vee (\neg i_2 \wedge \neg i_3) \vee (i_2 \wedge i_3)$

(b)

	Input vector or Condition $(i_1, \ldots, i_k)$	**Output value or Transition** (x)
F_0	(1,0,1)	0
	(1,1,0)	0
F_1	(0,0,0)	1
	(0,0,1)	1
	(0,1,0)	1
	(0,1,1)	1
	(1,0,0)	1
	(1,1,1)	1

$p = 6/8$

Figure 1.1: Three different ways of representing a Boolean function, shown for exemplar function of k=3 input variables: $x = f(i_1, i_2, i_3)$. **(a):** The logic formula specifying the possible conditions for which $x = 1$ (*True*) or $x = 0$ (*False*), according to Boolean algebra. **(b):** The Look-up table (LUT) of the function; left column lists all possible input vector conditions f_α, and the right column specifies the respective logical values assigned to automaton x, the output. The subsets of LUT entries that lead to the two logical values are denoted by F_0 and F_1 for $x = 0$ and $x = 1$, respectively; the output bias, p, of this function is listed at the bottom. **(c):** The Boolean hypercube is shown as a 3-dimensional cube, since $k = 3$; LUT entries are represented by the Euclidean coordinates of the corners of the cube, and the corresponding output values by the color of the circle embedded on each corner (black means *True* or 1 and white means *False* or 0).

1.3 Boolean networks

A Boolean network (BN) is a graph $\mathcal{B} = (X, E)$, where $X = \{x_i : i = 1, \ldots, n\}$ is a set of n Boolean automata *nodes* (or variables) and $E = \{e_{ji}\}$ is a set of *directed edges* where e_{ji} indicates an edge from x_j to x_i. The set of input nodes to x_i is denoted as $X_i = \{x_j \in X : e_{ji} \in E\}$. Its cardinality, $k_i = |X_i|$, is the *in-degree* of node x_i. At any given time t, $\mathcal{B}$ is in a specific *configuration* of node states, $\boldsymbol{x}^t = (x_1^t, \ldots, x_n^t)$. We use the terms *state* for individual automata (x) and *configuration* $(\boldsymbol{x})$ for the collection of states of the set of automata of $\mathcal{B}$, i.e. the collective network state. The state of a node x_i at time $t+1$ is determined by its LUT $F_i \equiv \{f_{i:\alpha} : \alpha = 1, \ldots, 2^k\}$, for every possible unique input vector $f_{i:\alpha}$ at time t. The output bias of node x_i is denoted as p_i.

Starting from an initial configuration, $\boldsymbol{x}^0$, a BN updates the states of its nodes with a *synchronous* or *asynchronous* policy. In this dissertation, the *synchronous update* policy is assumed, where the states of all nodes are simultaneously updated: $\boldsymbol{x}^{t+1} = (F_1(X_1^t), \ldots, F_n(X_n^t))$. The *dynamics* of $\mathcal{B}$ is thus defined by the temporal sequence of configurations that ensue, and there are 2^n possible configurations. The transitions between configurations can be represented as a *state-transition graph*, STG, where each vertex is a configuration, and each directed edge denotes a transition from $\boldsymbol{x}^t$ to $\boldsymbol{x}^{t+1}$. The STG of $\mathcal{B}$ thus encodes the network's entire *dynamical landscape.* Configurations that repeat, such that $\boldsymbol{x}^{t+\mu} = \boldsymbol{x}^t$, are known as *attractors*; *fixed point* when $\mu = 1$, and *limit cycle* – with period μ – when $\mu > 1$, respectively. The disconnected subgraphs of a STG leading to an attractor are known as *basins of attraction.* A BN $\mathcal{B}$ has a finite number b of attractors; each denoted by $\mathcal{A}_i : i = 1, ..., b$. Every configuration $\boldsymbol{x}$ is in the basin of attraction of some specific attractor $\mathcal{A}_i$. That is, the dynamic trajectory of $\boldsymbol{x}$ converges to $\mathcal{A}_i$. We denote such a dynamical trajectory by $\sigma(\boldsymbol{x}) \rightsquigarrow \mathcal{A}_i$.

BN are a type of discrete multivariate dynamical system introduced to build qualitative models of genetic regulation whereby attractors are thought of representing distinct phenotypic configurations (e.g. cell types) [65]. As we describe next, these dynamical systems are very amenable to large-scale statistical analysis (see [40] for a comprehensive review).

1.4 Boolean network ensembles

An *ensemble* of Boolean networks is a set of BNs characterized by some set of constraints; every member satisfies those constraints whose properties are otherwise randomized. A member of an ensemble shall be known as a *random Boolean network* (RBN). Stuart Kauffman has studied the nk ensemble in which every BN contains n nodes each containing k inputs; every other property including wiring makeup and the choice of Boolean functions is random [65–67]. Another well-studied ensemble is the nkp ensemble that requires, in addition to the nk constraint, that every node has a fixed output bias p. The original motivation behind the ensemble approach was to generate testable hypotheses about the interplay between self-organization and natural selection [67].

1.5 Perturbation spreading

An approach to characterize the behavior of a dynamical system is by way of studying its response to perturbations. One typically bases the characterization, e.g., robustness, stability or resilience, on the nature of short-term responses (see below). The study of long-term responses is more appropriate for models of spreading processes. In an epidemic setting, for instance, it's often important to understand the probability with which nodes at a specified distance from a source of infection could get affected [14, 34]. In a social setting, likewise, how ideas and opinions spread in the long run is an important question [9, 101]. In biology, spreading processes have found analogies in information diffusion in the brain [93] and even in the characterization of protein misfolding [59]. In systems biology, patterns of perturbation spreading hold the key to formulating strategies for intervention and damage mitigation in regulatory pathways [23, 116, 139]. In all these works, the research question essentially is: how do perturbations spread in the long run? — the exact adaptation of the term ‘perturbation’ is peculiar to the work (e.g., an infection in epidemics models is equivalent to a perturbation).

The short-term response to perturbations of a dynamical system, on the other hand, underlies the formulation of its dynamical stability. One is typically interested in the dynamical stability of the solutions of the system if they are available, or of a generic state if canonical solutions are not available, like in chaotic dynamical systems (e.g., Lyapunov exponent) [124]. The problem of stability is an important one since it has a bearing on the functioning of the system. In particular, real-world dynamical systems are generally not expected to be too unstable [67] – an intuitive expectation that led to the formulation of the criticality hypothesis (see below) [110]. There are other closely related notions of stability, namely robustness and resilience, which refer to the ability of a system to retain its basic functioning in the face of mutations to or failure of parts of the structure or dynamics of the system [31, 38]. The focus of this dissertation is rather on the aforementioned concept of dynamical stability.

The main question of this dissertation pertains to the characteristics of a system that are predictive of the perturbation-spreading behavior, whether in the long-term or short-term. As for long-term propgation, they are typically macro structural characteristics like modularity, density and connectedness [101, 121, 127]. It was recently found that certain characteristics of the transfer functions also are predictors on top of topological features [15]; the findings are, however, restricted to linear systems or non-linear systems with pair-wise interactions only (hence, Boolean networks are excluded). As for short-term propagation, the typical determinants are local features like mean degree, degree distribution, assortativity and output-bias [18, 73]. While the main focus of this dissertation is on the predictors of short-term perturbation-spreading in BN (chapters 2 and 3), the last chapter is dedicated to predicting long-term perturbation-spreading behavior. In both cases, the focus is specifically on the role of canalization as the predictor.

1.5.1 Criticality

The criticality hypothesis states that biological systems are *critical*, a condition that endows them with enough instability to reach other dynamical configurations, but not so much instability as to be sensitive to the tiniest perturbation [99, 110]. Further hypotheses have been put forth on the potential characteristics of critical systems which may have been selected for by evolution [110,127]. The criticality of a dynamical system is defined by the particular way in which small perturbations spread in the short-term, namely that they neither amplify nor dissipate.

This dissertation focuses on the criticality of Boolean networks (BN). As mentioned before, these discrete dynamical systems were originally introduced as models of gene regulation [67]. Many gene regulatory systems are known to be critical [10], and critical BNs have been shown to be most adaptive and evolvable [5]. Although, it has also been suggested that criticality is not a necessary condition for adaptability when it comes to BNs with scale-free structure [4]. A few biochemical systems are also known to be more ordered than critical [118]. An example is the segment polarity network of *Drosophila* which is known to be highy robust to perturbations [88, 136]. Our own analysis of a BN repository known as 'Cell collective' reveals a few other examples of biochemical systems that are more closely associated with the ordered regime (Fig. 2.13 in Ch.2).

Stuart Kauffman inquired into the extent to which self-organization[2] interacts with natural selection in the evolution of stable biochemical systems [65]. He hypothesized that a system with just a few inputs per component may be expected to be dynamically stable or critical, irrespective of the way it is connected, and that natural selection picked such systems whose stability came for "free",

---

[2] 'Self-organization' is a dynamical process by which the constituents of a complex system interact with each other, and settle down onto an attractor.

thus ascribing a ‘principle of least effort’ to evolution. A good example is a property of metabolic networks known as ‘molecular specificity’, wherein an enzyme could catalyze only one substrate and a substrate could be catalyzed by very few enzymes [98]. To test the hypothesis, Kauffman studied the dynamical properties of *nk* ensembles, which resulted in the first theories of criticality. A theory of criticality of Boolean networks would allow one to predict if a BN is stable, critical or chaotic based on some of its static properties like k and p. Kauffman’s goal was to generate new hypotheses about how natural selection may have picked certain ensembles that real biochemical systems might relate with — which however shall not be a subject of concern for this dissertation. The goal of this dissertation is rather to formulate better theories of criticality based on canalization in order to better characterize the dynamics of BN.

1.5.2 Measures of criticality

The *dynamical regime* of a BN can be classified into three categories: stable, critical or chaotic. A *Derrida map* (DM) is a recurrence plot relating the mean Hamming distances between pairs of configurations at time t and at $t+1$, with which the dynamical regime of a BN can be inferred [33]. Specifically, the slope-at-origin of the DM is an indicator of the dynamical regime, the particulars of which follow.

The Hamming distance between two configurations $\boldsymbol{x}$ and $\boldsymbol{y}$ of a BN is: $d(\boldsymbol{x}, \boldsymbol{y}) = \sum\limits_{i} |x_i - y_i|$. Let $\tilde{\mathbf{x}}^t$ denote a configuration resulting from a perturbation to $\boldsymbol{x}^t$, and $m = d(\boldsymbol{x}^0, \tilde{\mathbf{x}}^0)$ denote the size of the initial perturbation. The following quantity shall be used as a measure of how perturbations spread in a Boolean network:

$$D_m^t = \mathbf{E}\big[d(\boldsymbol{x}^t, \tilde{\mathbf{x}}^t) \,|\, d(\boldsymbol{x}^0, \tilde{\mathbf{x}}^0) = m\big] \tag{1.1}$$

where $\mathbf{E}$, the expected value, is taken over all possible pairs $(\boldsymbol{x}^0, \tilde{\mathbf{x}}^0)$ seperated by m for some $t \geq 0$.

The value of D_m^1 for some small integer m is known as the *Derrida coefficient* (D_c), which is an *order parameter* and a popular measure of stability developed for BNs [32, 33, 61, 67]. One may also consider a range of initial perturbation sizes $M = [m_1, m_2], 1 \leq m_1 \leq m_2 \leq N$ and a fixed interval of time $t \in [1, T]$, then calculate $D_{m_i}^1$ for each m_i and t, and summarize them in some way. Varying the range of initial perturbation sizes allows us to study the collective behavior of the BN across various scales, and varying the time interval allows us to study long-term behavior of perturbation-spreading. In this dissertation, the following normalized form of D incorporating both the M and t parameters, shall be used to characterize perturbation-spreading:

$$\widehat{D}_M^t = \frac{\sum\limits_{m_1 \leq m_i \leq m_2} D_{m_i}^t}{\sum\limits_{m_1 \leq m_i \leq m_2} m_i} \tag{1.2}$$

for a given $M = [m_1, m_2]$ and every $t \in [1, T]$. Hence, $\widehat{D}_M^0 = 1$ for any M, that is, the normalized initial perturbation size is equal to 1 regardless of the actual perturbation sizes.

The D_c is a special case of the above quantity, with $t = 1$, that is:

$$\begin{aligned} & D_c = \widehat{D}_M^1 \\ & \text{where } M = [1, n/10]. \end{aligned} \tag{1.3}$$

The dynamical regime of a BN is inferred based on the value of D_c as follows:

$$\begin{aligned} & \text{Ordered, if: } D_c < 1, \\ & \text{Critical, if: } D_c = 1, \text{ and} \\ & \text{Chaotic, if: } D_c > 1. \end{aligned} \tag{1.4}$$

As noted, the range of initial perturbations sweeps a small percentage of the size N of the BN.

Thus, the D_c provides a measure of the rate of spread of perturbations at $t = 0$. Specifically, as the definition above suggests, the D_c is the slope at the origin of the DM. This is a sufficient characteristic of dynamical regime because, for RBNs with homogenous connectivity (like the ones in the *nkp* ensembles), the values of D^1_m is a monotonically increasing function of m [10]. An example of an alternative order parameter for random BNs with homogeneous connectivity is the 'Lyapunov exponent' [82], that explicitly relies on the characteristic k and p of the BN. This order parameter is not suitable for our purpose since the BNs we consider are characterized by parameters other than k.

1.5.3 Theories of criticality

A 'theory of criticality' is a condition that must be satisfied by the parameters of an RBN ensemble, also known as the *tuning parameters*, so that it is dynamically critical on an average. That is, it's an ensemble-based theory. In the following, we shall denote the mean in-degree of the BN $\langle k \rangle$ by k, and the mean output bias $\langle p \rangle$ by p, unless stated otherwise. That is, the k and p are the same for the automata in a BN. The subscript 'c' means "critical value".

The very first theory of criticality of Boolean networks was put forward by Kauffman from his studies of *nk* ensembles in which n was swept across a range of values, and $k \in [1, n]$ represented the fixed number of inputs every node received [67]. His simulations showed that the criticality condition is given by[3]:

$$k_c = 2 \tag{1.5}$$

There was some initial evidence from real gene regulatory systems and metabolic networks suggest-

---

[3] Analytically derived by Derrida using a technique called 'annealed approximation' where during each update-step the connectivity and transfer functions are randomized under parameter constraints [32].

ing that they related with the $k = 2$ ensemble [67, 68]. Moreover, many of the attractor statistics of the $k = 2$ ensemble seemed to match with those of real systems. For instance, the number of attractors matched the number of cell types given different values of n representing genome length [67]. However, it is now known that many real genetic networks have many more than two inputs on the average; it was estimated to be close to five even at a time when comparitively little data was available [6]. For example, the *E.coli* gene transcription network nodes have up to ten inputs [10]. These observations beg the question of whether k is a meaningful characteristic.

A more accurate criticality condition was derived by Derrida et al. [32] that allowed systems with $k > 2$ to be stable through appropriate tuning of p. In other words, the improved theory implies that $k = 2$ is not a necessary condition for criticality; it is still a roughly sufficient condition since the previous theory is not invalidated. The more nuanced criticality condition is given by (illustrated in fig 1.2):

$$2k_c p(1-p) = 1 \tag{1.6}$$

Although derived from the assumption that k is the same for all nodes (drawn from a Delta distribution), it is known that the above criticality condition holds as long as k is a valid first moment of some distribution [4,37]. Nevertheless, this theory still prescribes an unrealistic necessary condition: it requires that P be very low or very high for large values of k. For $k = 10$, for example, the above expression requires that $p \geq 0.95$ or $p \leq 0.05$ for the ensemble to be on the order side of the boundary, which real biochemical systems don't always satisfy [4]. For example, the average p in the *E.coli* transcription network is about 0.576 [10].

It is common for real systems, however, to possess heterogeneous in-degree distributions (e.g.,

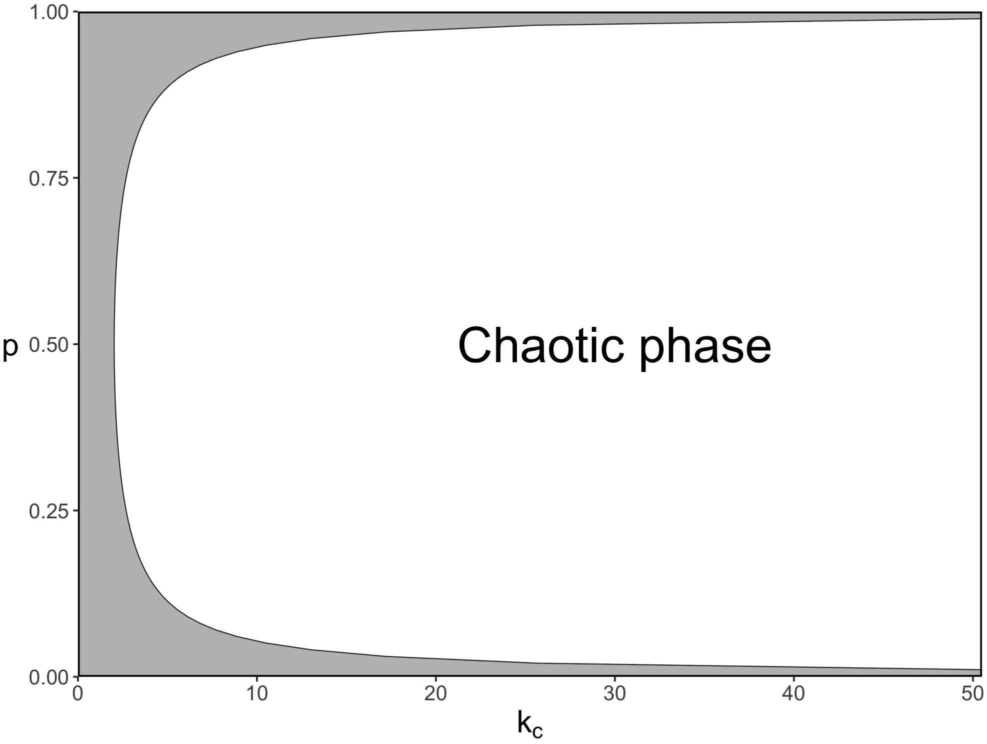


Figure 1.2: The phase diagram of RBNs in the space of k and p.

power-law). Due to extreme heterogeneity, however, the mean[4] $\langle k \rangle$ (signal) of the distribution could lose its relevance due to a relatively large variance (noise) for large networks. For example, consider an in-degree distribution governed by power-law, given by the zeta distribution:

$$\begin{aligned} &p_\gamma(k) = k^{-\gamma}/\zeta(\gamma) \text{ where,} \\ &\zeta(\gamma) = \sum_{n=1}^{\infty} n^{-\gamma} \end{aligned} \tag{1.7}$$

When $\gamma \in (2, 3]$, the variance is infinity. Consequently, Aldana considered the power-law exponent γ itself as a tuning parameter in place of k, and derived a new criticality condition in the γ-p space [4] (fig 1.3):

---

[4] In heterogeneous networks, k is obviously not the same for all automata, so we use the notation $\langle k \rangle$ instead of k

$$
\begin{aligned}
&2p(1-p)\frac{\zeta(\gamma_c - 1)}{\zeta(\gamma_c)} = 1 \text{ where,}\\
&\langle k \rangle = \frac{\zeta(\gamma_c - 1)}{\zeta(\gamma_c)} \text{ (mean in-degree) exists for } \gamma_c > 2. \qquad (1.8)\\
&\text{When } \gamma_c \leq 2,\ \langle k \rangle = \infty.
\end{aligned}
$$

That is, when the scale-free exponent $\gamma_c \leq 2$, networks are always chaotic, essentially because $\langle k \rangle = \infty$. When $\gamma_c \in (2, 3]$, with a finite $\langle k \rangle$ and an infinite variance (blurring the relevance of the mean), a phase transition exists and depends on p. With $\gamma_c > 3$, the networks are always in the ordered regime. Numerical simulations, however, show that for finite-sized networks, the critical boundary is not as sharp [77].

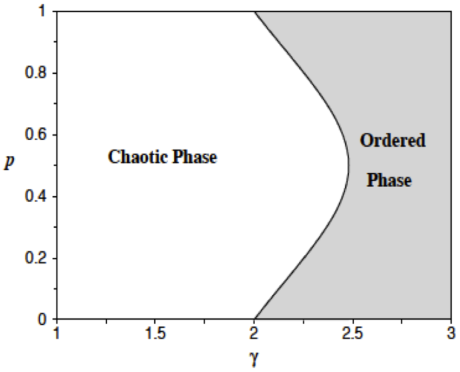


Figure 1.3: (Picture source: Ref. [4])The phase diagram of RBNs with scale-free in-degree distribution in the space of γ and p.

Lee and Reiger studied structural heterogeneity more generally by considering a generic joint probability distribution of in- and out- degrees [77]. Their analytically derived criticality equation is a generalization of equation (1.6):

$$\frac{\langle k^{in} k^{out} \rangle}{\langle k \rangle} = \frac{1}{2p(1-p)} \qquad (1.9)$$

The left hand side of the above equation can be approximated by the largest eigenvalue, λ, of the adjacency matrix of the BN [109], leading to an alternative criticality condition:

$$\lambda_c = \frac{1}{2p(1-p)} \tag{1.10}$$

It is well know [30] that $\lambda \propto \langle k \rangle$. It is also known from the work in [109] that λ could vary even for a fixed value of $\langle k \rangle$. Thus, λ is one way of quantifying "pure" structural heterogeneity (without any confounding from $\langle k \rangle$) that is able to determine criticality. More work has been done along these lines, for example, reference [106] studies the interaction of λ with transfer function characteristics like sensitivity (described below) in determining criticality, and the role of network assortativity in influencing the same.

Thus, we can see that heterogeneity in k plays a role in criticality. However, in this dissertation we only consider homogeneous networks, particularly where k is the same for all automata as mentioned above. We plan to consider heterogeneous networks in future work.

1.6 Canalization in Boolean automata

The notion of *biological canalization* was proposed independently by Waddington and Schmalhausen as a mechanism that links micro-evolution (genotype) to macro-evolution (phenotype) [123]. Two different notions of canalization exist: 'genetic canalization' wherein organisms with different genetic makeup have the same phenotype, and 'environmental canalization' wherein organisms with the same genetic makeup when exposed to different environments that could epistatically influence their gene expressions have the same phenotype [36, 42, 123]. In this dissertation, we refer to the latter version which is often identified with the buffering of developmental processes. Although canalization has been demonstrated in nature (e.g., in the fruit fly *Drosophila* and in the flowering plant *Arabidopsis* [107, 111]), it is an open question as to whether canalization is naturally selected for or if it is a byproduct of selection of other features [123].

The mathematical analogue of biological canalization is known as *logic canalization* which was originally defined as a property of certain Boolean functions by which the state of a single input variable fully determines the output [67], rendering other input variables redundant. Many gene regulatory systems have been reported to possess highly canalizing functions [10, 48, 49, 67, 127]. Thus, canalization is thought to play an influential role in the dynamics, particularly criticality, of biological systems. From now on, unless otherwise specified, when we refer to canalization we mean logic canalization.

1.6.1 Types of canalization

The notion of canalization was originally conceived by Stuart Kauffman [67], which was later refined and generalized by various groups. Broadly speaking, canalization comprises of the following three categories: (1) canalization (the original conception), (2) nested canalization and (3) partial canalization.

Original conception

A Boolean function is a *canalizing function* if it contains at least one *canalizing input variable* and a corresponding *canalizing input value* that fully determines the *canalized output value* of the function [64]. Formally stated, a Boolean function $x = f(i_1, \ldots, i_k)$ is a canalizing function iff:

$$\exists i_j \in \{i_1, \ldots, i_k\} \text{ and } a, b \in \{0, 1\} : i_j = a \implies x = b \tag{1.11}$$

For example, the logical OR function $x = i_1 \vee i_2$ is a canalizing function because,

$$i_1 = 1 \implies x = 1$$

$$i_2 = 1 \implies x = 1$$

Whereas, the logical XOR function $x = i_1 \oplus i_2$ is *not* a canalizing function because it does not have a single canalizing input variable (both inputs are required):

$$(i_1 = 0 \wedge i_2 = 1) \implies x = 1$$

$$(i_1 = 1 \wedge i_2 = 0) \implies x = 1$$

It has been shown that criticality depends on the proportion of canalizing functions in a BN [62]. A recent work also shows that correlations between in-degree and canalization influences network stability. In particular, a strong correlation between high in-degree and parity functions (XOR-like) promotes instability, while a combination of low in-degree and OR/AND-like functions promotes stability [122].

Nested canalization

Nested canalization occurs when a given input variable is canalizing when a sequence of other canalizing input variables do *not* assume their canalizing input values [63]. The set of nested canalizing functions is naturally a subset of the set of canalizing functions. Stated formally, a Boolean function $x = f(i_1, \ldots, X_k)$ is a *nested canalizing function* (NCF) iff:

$$\begin{aligned}
&\exists(j_1, \ldots j_k) \in \sigma(I) \text{ and } a_1, \ldots a_k, b_1, \ldots b_{k+1} \in \{0,1\} : \\
&(j_1 = a_1 \implies x = b_1) \wedge \\
&\vdots \\
&((j_1 \neq a_1 \wedge \neq \wedge \ldots j_k \neq a_k) \implies x = b_{k+1})
\end{aligned} \tag{1.12}$$

where, $I = \{i_1, \ldots, i_k\}$ and $\sigma(I)$ denote the set of permutations of I.

An example NCF with $k = 3$ is shown in Fig. 1.4.

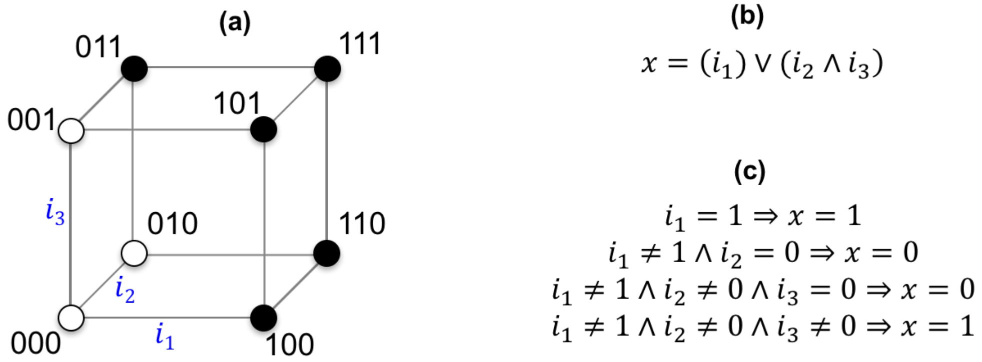


Figure 1.4: An example NCF Boolean function. **(a):** The hypercube of the BF. **(b):** The logic formula of the BF **(c):** The hierarchical nature of the BF that makes it an NCF.

As NCFs are quite sparse in the space of all Boolean functions, they were brought under a broader umbrella of *partially nested canalizing functions* (PCNF) with the aid of a 'canalizing depth' measure [75]. According to this measure, NCFs have full depth k, non-canalizing functions have depth 0, all other functions with at least one canalizing input variable have an intermediate depth. It was found that greater canalizing depth renders the BN with more stability, even though the gain in stability rapidly drops with depth [75]. Similar results were found in a study involving a subset of PCNFs [60]. The 'layer number' of a NCF is analogous to its canalizing depth and has also been found to reliably influence network stability [78].

It has been reported [63, 64] that many automata found in the Systems biology models literature are NCF. However, the proportion of canalizing functions, let alone NCFs, dramatically reduces with k, which is in the order of 10^{-9} for $k = 6$ [62]. This fact implies that the role of canalization would have to be minimal in networks with large connectivity. This apparent relegation of the role of canalization to the margin is, however, limited only by its somewhat narrow definition, as we shall see next.

Partial canalization

A more general definition of canalization is known as *partial canalization* that allows *subsets* of input variables, rather than single variables, to collectively canalize the automata output [108]. This definition capitalizes on the fact that almost every Boolean function contains canalizing subsets of input variables as opposed to single variables [108]. Stated formally, a Boolean function $x = f(i_1, \ldots, i_k)$ is a *partially canalizing* function iff:

$$\begin{aligned} &\exists E = \{i_l, i_m, \ldots\} \subset I = \{i_1, \ldots, i_k\} \text{ and } a_l, a_m, \ldots\, b \in \{0, 1\} : \\ &((i_l = a_l) \wedge (i_m = a_m) \wedge \ldots) \implies x = b \end{aligned} \tag{1.13}$$

where, E is a canalizing set of input variables, and $\{a_l, a_m, \ldots\}$ the corresponding set of input values. Reichhardt and Bassler [108] have shown that this more general class of partially canalizing functions dominates the space of Boolean functions for any number of inputs k. Indeed, for any value of k, there are only two *non-canalizing* functions that always depend on the states of all inputs. For $k = 2$, the two non-canalizing functions are XOR and its complement XNOR; for $k > 2$, they are the parity functions. So, for all but the parity functions, there is always a subset E of input variables, such that $|E| < k$, which controls the automata output state when in some specific state $\{a_l, a_m, \ldots\}$. Naturally, $E \equiv \emptyset$ for constant functions. The set of partially canalizing functions is thus a superset of the set of canalizing functions and very common.

1.6.2 Schema redescription

The *schema redescription* method was introduced for Cellular Automata [86, 87, 89] and Boolean network models in Systems Biology [88]. It is an approach to systematically capture and quantify partial canalization in Boolean automata. Following is a summary of the concepts and definitions relevant to this dissertation.

The core concept of this framework is that of *input schema*. An input schema, f'_v, is exactly like an LUT entry f_α (§1.2), except it allows an additional *wildcard* symbol '#' to appear in its condition. When the state of input variable i is given by wildcard symbol, it means that its logical value does not affect the output of automaton x.

That is:

$$\begin{aligned}&\text{if } i_j = \#, \text{ then}\\&f(i_1, \ldots, i_j, \ldots, i_k) = f(i_1, \ldots, \neg i_j, \ldots, i_k)\end{aligned} \tag{1.14}$$

It is a concept directly related to the definition of partial canalization Defn. (1.13). The non-wildcard input variables (literals) $e \in E$ in an input schema are referred to as *effective inputs*, or *enputs* for short. The enputs of a schema are sufficient to determine the output, while the remaining input variables $i \in I \wedge i \notin E$ are *redundant inputs* for that schema's condition.

As a consequence of Defn. 1.14, an input schema *redescribes* a subset of LUT entries: $\Upsilon_v \equiv \{f_\alpha : f_\alpha \rightarrowtail f'_v\} \subseteq F$ ($\rightarrowtail$ means 'is redescribed by'). For example, the input schema $f'_v = (0, 1, \#)$ redescribes the set of LUT entries $\Upsilon_v \equiv \{(0, 1, 0), (0, 1, 1)\}$. The set of all input schemata that redescribes the entire LUT is denoted as: $F' \equiv \{f'_v\}$. Note that it is possible that there are LUT entries that are redescribed by more than one input schemata (see Fig. 1.6 for an example).

There are typically many possible sets F' that completely redescribes LUT F. However, there is a unique F' that shall always be considered as the representative set for any given LUT, which satisfies the following properties:

1. F' is *minimal.* This means that none of the non-wildcard inputs in the condition of any schema in F' can be 'raised' to the wildcard status and still ensure that the automaton transitions to the same state. That is, $(\Upsilon_v \nsubseteq \Upsilon_\phi) \wedge (\Upsilon_\phi \nsubseteq \Upsilon_v), \forall f'_v, f'_\phi \in F'$.
2. F' is *complete.* This means that for a given LUT F there is a unique F' that contains all possible minimal input schemata.

The subsets of input schemata that go to outputs 0 and 1 are respectively denoted as F'_0 and F'_1, so $F' \equiv F'_0 \cup F'_1$, where $F'_0 \cap F'_1 = \phi$.

Each wildcard schema in F'_b, where $b \in \{0, 1\}$, represents a conjunctive clause, and the set F'_b as a whole represents a disjunctive clause (§ 1.2). In other words, each wildcard schemata set is equivalent to a logic formula (Fig. 1.6). For example, $F'_1 = \{(0, \#, 1), (0, 1, \#)\}$ associated with some automaton x with inputs i_1, i_2, i_3 represents the logic formula: $x = (\neg i_1 \wedge i_3) \vee (\neg i_1 \wedge i_2)$. The complementary set $F'_0 = \{(\#, 0, 0), (1, \#, \#)\}$ associated with the same automaton x represents the complementary logic formula: $\neg x = (\neg i_2 \wedge \neg i_3) \vee (i_1)$. We use the notation F'_1 to represent conventional logic formulas, since they usually have an uncomplemented variable on their left-hand sides.

We use an additional subscript within parentheses to refer to the automaton with which the schemata set is associated in a BN: $F'_{0(i)}$ and $F'_{1(i)}$ are associated with automaton x_i.

The essence of the schema redescription method is that it *compresses* the sets F_0 and F_1 into the sets F'_0 and F'_1 respectively. That is, $|F'_b| \leq |F_b, \forall b \in \{0, 1\}$. In summary, partial canalization is identified with the compression achieved by the schema redescription process; the more canalized automaton x is, the larger the compression of its LUT: $|F|/|F'|$.

The input schemata in F' are equivalent to the prime implicants obtained in the first step of Quine-McCluskey's method of logic minimization, the tabular method [91], which is widely used in the simplification of digital circuits.

The *two-symbol schema redescription* method extends the above method further by redescribing sets of input schemata. A two-symbol input schema, f''_θ, is exactly like an input schema, except it allows an additional *position-free* symbol $\dot{i}$ (notice the dot above i) to appear in its condition. Any subset of inputs thus marked can permute among themselves in all possible ways while leaving the transition invariant. We refer to such subsets as *group-invariant enputs*—that, moreover, may include wildcard symbols marked with a position-free symbol. A two-symbol schema, f''_θ redescribes a subset $\Theta'_\theta \equiv \{f'_\upsilon : f'_\upsilon \rightarrowtail f''_\theta\}$ of input schemata; it naturally also redescribes a subset $\Theta_\theta \equiv \{f_\alpha : f_\alpha \rightarrowtail f''_\theta\}$ of LUT entries. For example, the two-symbol input schema $f''_\theta = (1, \#, \dot{0}, \dot{\#})$ redescribes the subset $\Theta'_\theta \equiv \{(1,\#,0,\#),(1,\#,\#,0)\}$ of input schemata and the subset $\Theta_\theta \equiv \{(1,0,0,0),(1,0,0,1),(1,0,1,0),(1,1,0,0),(1,1,0,1),(1,1,1,0)\}$ of LUT entries. The set of all two-symbol input schemata that redescribes the entire LUT is denoted as: $F'' \equiv \{f''_\theta\}$. The subsets of two-symbol input schemata that go to outputs 0 and 1 are respectively denoted as F''_0 and F''_1, so $F'' \equiv F''_0 \cup F''_1$, where $F''_0 \cup F''_1 = \phi$. An LUT entry can be redescribed by more than one two-symbol input schema (see Fig. 1.8 for an example).

The LUT entry redescribed by a two-symbol schema must satisfy the following condition:

$$\bigwedge_{i_j \in X_l^0} \neg i_j \bigwedge_{i_j \in X_l^1} i_j \bigwedge \left(n_g^0 \leq \sum_{i_j \in X_g} \neg i_j \leq n_g^0 + n_g^{\#} \right) \bigwedge \left(n_g^1 \leq \sum_{i_j \in X_g} i_j \leq n_g^1 + n_g^{\#} \right)$$

where,

$f''_\theta = (i_1, \ldots, i_k)$ represent a two-symbol input schema, and let

$X_g = \{i_j : i_j \in \{\dot{0}, \dot{1}, \dot{\#}\}\}$ denote the set of group-invariant enputs, (1.15)

$X_l = \{i_j : i_j \in \{0, 1\}\},$ denote the set of literal inputs, and

$X_g^0 = \{i_j : i_j \in X_g \wedge i_j = \dot{0}\},\ X_g^1 = \{i_j : i_j \in X_g \wedge i_j = \dot{1}\},$ and $X_g^\# = \{i_j : i_j \in X_g \wedge i_j = \dot{\#}\},$ and

$X_l^0 = \{i_j : i_j \in X_l \wedge i_j = 0\},\ X_l^1 = \{i_j : i_j \in X_l \wedge i_j = 1\},$ and

$n_g^0 = |X_g^0|,\ n_g^1 = |X_g^1|,$ and $n_g^\# = |X_g^\#|.$

In summary, the two-symbol schema redescription method identifies additional compression, if any, beyond what the original schema redescription achieves. That is, $|F''| \leq |F'|$. This additional compression derives from permutation symmetries that exist in the input variables of automata, and we can think of it as an additional form of redundancy associated with canalization [88]

Indeed, an overall point to be gleaned from the above two methods of schema-redescription is that each method progressively identifies more canalization and thus progressively compresses the Boolean hypercube. This point is visually illustrated in Fig. 1.5. The two types of redundancy removed via schema redescription can be measured [88] as described in the next subsections.

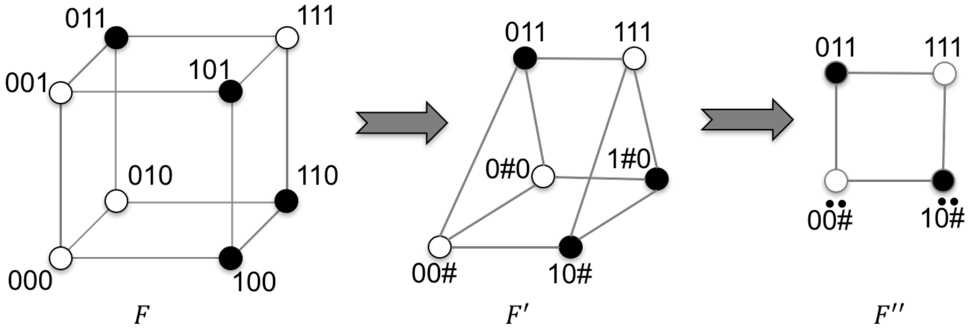


Figure 1.5: A Boolean function is progressively compressed by schema-redescription. Two schemata are connected by an edge if there is at least one LUT entry in the corresponding redescribed sets, which are separated by a Hamming distance of 1 bit.

1.6.3 Effective connectivity

Wildcard schemata identify redundant inputs in the LUT entries of an automaton x, which are marked with the wildcard symbol. Therefore, this form of redundancy can be easily measured by tallying the mean number of redundant input variables in F':

$$k_r(x) = \frac{\sum\limits_{f_\alpha \in F} \max\limits_{v: f_\alpha \in \Upsilon_v} (n_v^{\#})}{|F|} \tag{1.16}$$

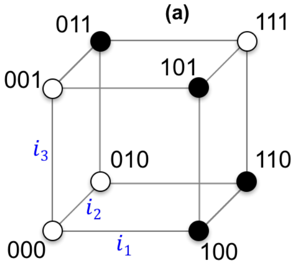
where $n_v^{\#}$ is the number of wildcards in schema f'_v that redescribes the subset of LUT entries f_α, Υ_v, and the max is taken over all the schemata redescribing f_α for each f_α. Naturally, the more canalized x is, the larger its *input redundancy* $k_r(x)$ is, because LUT entries tend to be redescribed with schemata with many wildcards.

A complementary measure of input redundancy is defined as the *effective connectivity* of an automaton x [88]:

$$k_e(x) = k - k_r(x) \tag{1.17}$$

which is the mean number of effective input variables of x, assuming all LUT entries are equally likely.

A detailed example of the calculation is show in Fig. 1.6.



(b)

$$x = (\neg i_1 \wedge i_2 \wedge i_3) \vee (i_1 \wedge \neg i_2) \vee (i_1 \wedge \neg i_3)$$

$$F'_1 = \{(0,1,1), (1,0,\#), (1,\#,0)\}$$
$$F'_0 = \{(1,1,1), (0,0,\#), (0,\#,0)\}$$

(c)

f_α	x	Redescribing f'_v	$\max(n_v^{\#})$
(0,0,0)	0	{(0,0,#), (0,#,0)}	1
(0,0,1)	0	{(0,0,#)}	1
(0,1,0)	0	{(0,#,0)}	1
(1,1,1)	0	{(1,1,1)}	0
(0,1,1)	1	{(0,1,1)}	0
(1,0,0)	1	{(1,0,#), (1,#,0)}	1
(1,0,1)	1	{(1,0,#)}	1
(1,1,0)	1	{(1,#,0)}	1

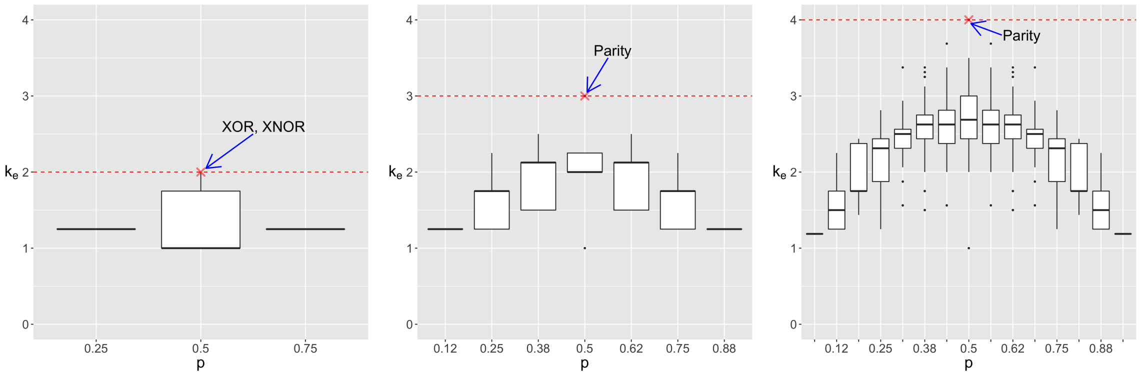
(d)

$$k_r(f) = {}^6\!/_8 = 0.75$$
$$k_e(f) = k - k_r(f) = 2.25$$

Figure 1.6: Computing the effective connectivity of a Boolean automaton x, defined by a Boolean function of k=3 input variables: $x = f(i_1, i_2, i_3)$. **(a):** The k-dimensional hypercube representation of f. **(b):** The logic formula and the sets of redescribing input schemata. There is a one-to-one correspondence between the prime implicants and the schemata going to 1. Every input schema is represented by a corresponding 'subcube' in (a). **(c):** Overlapping sets of redescribing input schemata and the maximum number of wildcard symbols per LUT entry. Notice that some LUT entries are redescribed by more than one input schemata. **(d)**: Calculation of k_r whose numerator is the total of the last column in (c) and k_e.

In this formulation of canalization, the conventional canalizing functions described in the previous section have the smallest possible k_e (maximal k_r), the non-canalizing functions have maximal k_e, and every other function has an intermediate value. In this sense, k_e can be thought of as an *effective* in-degree of a Boolean function that can assume fractional values in the range $[0, k]$: the average number of input variables that actually control the automaton's output.

The distribution of k_e for all possible non-constant Boolean functions for $p \in [0,1]$ and for $k \in \{2,3,4\}$ is shown in Fig. 1.7. The important point here is that when p and k—the "apparent" connectivity—are fixed, the effective connectivity can still vary depending on the actual logic of the automata also manifested in the structure of the hypercubes. Fig. 2.2 in § 2.2.1 shows similar plots for sampled spans of Boolean functions with $k \in \{6,8\}$.



(a) $k = 2$ **(b)** $k = 3$ **(c)** $k = 4$

Figure 1.7: The distribution of k_e for all possible non-constant functions for every $p \in (0,1)$ and for every $k \in \{2,3,4\}$. The horizontal red line represents $\max(k_e) = k$ for each p, and the associated parity functions in each case are marked.

A related measure known as 'Boolean complexity'—introduced in a psychology study—counts the total number of 'literals' in the simplified DNF of a Boolean function [35], which is the same as the total number of effective inputs of each input schema in F'. The difference is that Boolean complexity does not account for overlaps among the input schemata that comprise F', while k_e does.

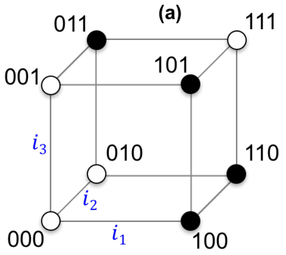
1.6.4 Input symmetry

Two-symbol schemata, in addition to redundant inputs, also identify symmetric inputs of an automaton x, which are marked with the position-free symbol. Therefore, this additional form of redundancy can be easily measured by tallying the mean number of permuting input variables in

F'':

$$k_s(x) = \frac{\sum\limits_{f_\alpha \in F} \max\limits_{\theta : f_\alpha \in \Theta_\theta} (n_\theta^\bullet)}{|F|} \tag{1.18}$$

where $n_\theta^\bullet$ is the number of position-free symbols in the schema f''_θ that redescribes Θ_θ, and the max is taken over all the two-symbol schemata redescribing each LUT entry f_α. Similarly to input redundancy, the more symmetry x has, the larger its *input symmetry* $k_s(x)$, because LUT entries tend to be redescribed with schemata with many poistion-free symbols. A detailed example of the calculation is show in Fig. 1.8.



(b)

$x = (\neg i_1 \wedge i_2 \wedge i_3) \vee (i_1 \wedge \neg i_2) \vee (i_1 \wedge \neg i_3)$

$F''_1 = \{(0,1,1), (1, \dot{0}, \dot{\#})\}$

$F''_0 = \{(1,1,1), (0, \dot{0}, \dot{\#})\}$

(c)

f_α	x	Redescribing f''_θ	$\max(n_\theta^\bullet)$
(0,0,0)	0	$\{(0, \dot{0}, \dot{\#})\}$	2
(0,0,1)	0	$\{(0, \dot{0}, \dot{\#})\}$	2
(0,1,0)	0	$\{(0, \dot{0}, \dot{\#})\}$	2
(1,1,1)	0	{(1,1,1)}	0
(0,1,1)	1	{(0,1,1)}	0
(1,0,0)	1	$\{(1, \dot{0}, \dot{\#})\}$	2
(1,0,1)	1	$\{(1, \dot{0}, \dot{\#})\}$	2
(1,1,0)	1	$\{(1, \dot{0}, \dot{\#})\}$	2

(d)

$k_s(f) = {}^{12}/_{8} = 1.5$

Figure 1.8: Computing the input symmetry of a Boolean automaton x defined by the Boolean function of Fig. 1.6. **(a):** The hypercube representation of f. **(b):** The logic formula and the sets of redescribing two-symbol schemata. Every two-symbol schema is represented by a corresponding a 'fractional subcube' (comprising a set of subcubes) in (a). **(c):** Sets of redescribing two-symbol schemata and the maximum number of position-free symbols per LUT entry. **(d)**: Calculation of k_s whose numerator is the total of the last column in (c).

In this formulation, majority functions have maximal k_s. Consider the majority function with $k = 3$

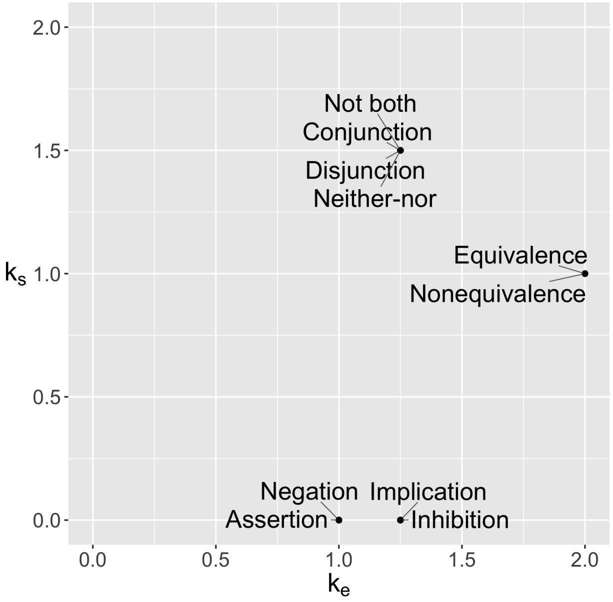
inputs, for example. The F'' for this function consists of: $F_0'' = \{(\dot{0}, \dot{0}, \dot{\#})\}$ and $F_1'' = \{(\dot{1}, \dot{1}, \dot{\#})\}$; since all LUT entries all redescribed by two-symbol schemata where input variables permute, $k_s = 3$.

In previous work by Marques-Pita and Rocha, a clear case of the relevance of input symmetry was demonstrated [87]. There, two Cellular automata rules that were evolved to perform the density classification task were found to differ only in their k_s. However, each rule leads to very distinct dynamical behavior, and the one with higher k_s was found to yield a slightly better performance, albeit in a statistically significantly way.

Figs.1.9 and 1.10 locate a few known and well-studied Boolean functions in the joint k_e-k_s space.

Function	Logic formulas
Disjunction	$i_1 \vee i_2$
Conjunction	$i_1 \wedge i_2$
Neither−nor	$\neg(i_1 \vee i_2)$
Not both	$\neg(i_1 \wedge i_2)$
Equivalence	$(i_1 \wedge i_2) \vee (\neg i_1 \wedge \neg i_2)$
Nonequivalence	$(i_1 \wedge \neg i_2) \vee (\neg i_1 \wedge i_2)$
Implication	$\neg i_1 \vee i_2,\ \ i_1 \vee \neg i_2$
Inhibition	$\neg i_1 \wedge i_2,\ \ i_1 \wedge \neg i_2$
Assertion	$i_1,\ i_2$
Negation	$\neg i_1, \neg i_2$

(a)



(b)

Figure 1.9: The k_e and k_s of all possible non-constant $k = 2$ functions. **(a)** The names of the functions and the corresponding logic formulas. The last four functions have two associated logic formuals each. **(b)** The k_e and k_s values. The function names suggest to some extent how canalizing or symmetric they may be. For example, 'inhibition' suggests one variable inhibits another, thus it may be canalizing. Likewise, 'neither-nor' and 'nonequivalence' suggest nothing special about one variable over another, hence they may be symmetric.

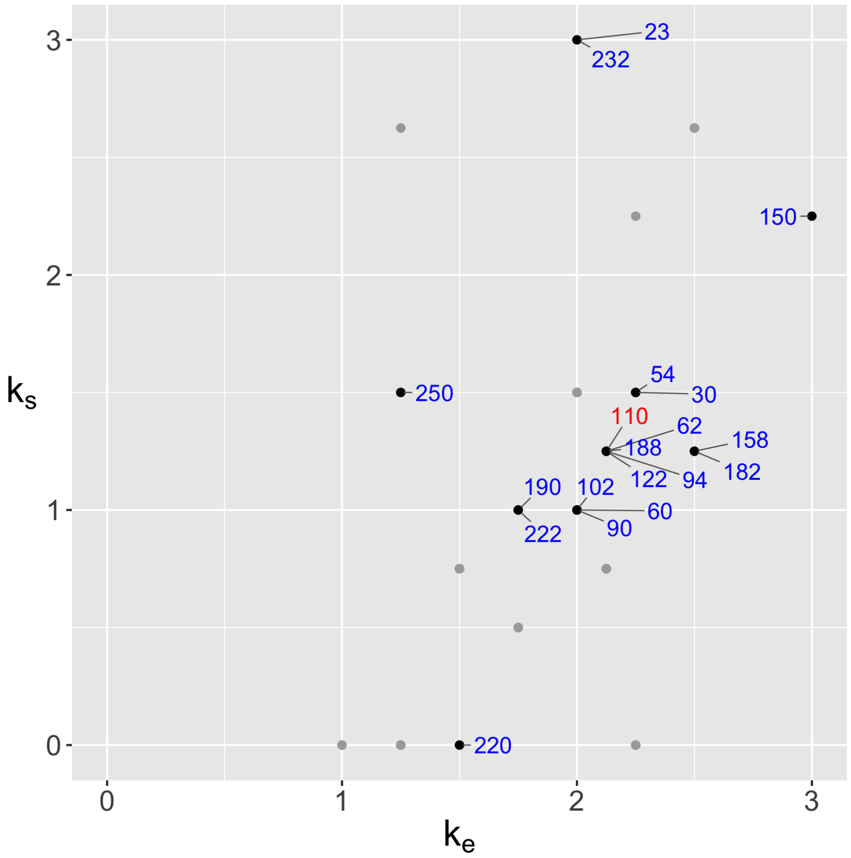


Figure 1.10: The k_e and k_s of all possible non-constant $k = 3$ functions, also known as 'elementary cellular automata' (ECA) rules. Each rule is identified by a decimal number equivalent of its output vector. A few well-studied rules are highlighted in black. Rule 110 (marked with a red label) known for its 'universal computation' properties has somewhat a high k_e and k_s suggesting that they may be necessary (although clearly not sufficient, since it is grouped with other rules namely 62,94,122,188) for universal computation. Rules 23 and 232 with maximal $k_s = 3$ are the majority function and its complement respectively.

1.6.5 Sensitivity

The sensitivity of a Boolean automaton is a probabilistic measure of canalization. The notions of *activity* and *sensitivity* of Boolean functions were introduced in Ref. [117] as measures of the stability of response to perturbations to inputs. The *activity*, a_j, of input variable i_j indicates the probability, $\Pr(\tilde{x} \mid \tilde{i_j})$, that x flips its state at $t+1$ if the input i_j flips its state at t, given an uniform distribution of input states at t. Formally stated,

$$\begin{aligned} a_j = \Pr\big(f(i_1, \ldots, i_j, \ldots, i_k) = 1, f(i_1, \ldots, \tilde{i_j}, \ldots, i_k) = 0\big) \\ + \Pr\big(f(i_1, \ldots, i_j, \ldots, i_k) = 0, f(i_1, \ldots, \tilde{i_j}, \ldots, i_k) = 1\big) \end{aligned} \tag{1.19}$$

The mean sensitivity, $s(x)$, is the sum of activities of its input variables. In other words, it's the expected number of inputs to which the automaton is sensitive to. Formally stated,

$$\begin{aligned} s(x) &= \sum_j a_j \\ &= \frac{\sum_j a_j}{k} \times k \\ &= \langle a \rangle \times k \end{aligned} \tag{1.20}$$

where, $\langle a \rangle$ denotes the mean input activity—the expected probability by which x flips if one of its inputs flips.

Fig. 1.11 describes a worked-out example. The smaller the value of $s(x)$, the less sensitive x is to random input perturbations because fewer variables actually influence the output. Hence, s is also a measure of canalization. The main difference between k_e and s is that the former quantifies the average number of input variables that *jointly* determine the output state, while the latter quantifies the average number of inputs that flip the output when *individually* flipped. Thus, k_e is a more natural characterization of partial canalization as defined above (§1.6.1), as it deals directly with joint canalization by subsets of inputs. In this sense, k_e can be viewed as a nonlinear characteristic, and s a linear characteristic of a Boolean function.

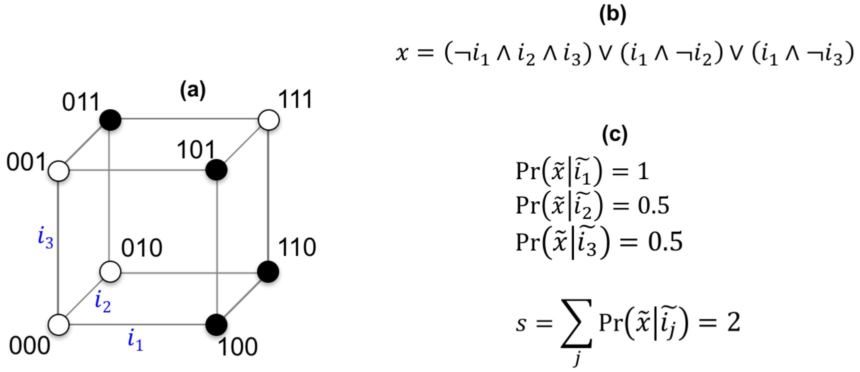


Figure 1.11: Calculating the mean sensitivity s of the same BF as in Figs. 1.6 and 1.8.

1.7 Canalization in Boolean network dynamics

A Boolean network itself is a vector-valued Boolean function: $\{0,1\}^n \mapsto \{0,1\}^n$. Thus, the notion of schema-redescription can naturally be extended to the BN. A schema, in this case, would group together sets of configurations that are indistinguishable in some way — say, they all go to the same configuration in the next update step or converge to the same attractor. Hence, the notion of micro-level canalization, which is the property of a single node (effective connectivity and input symmetry), can be extended into 'macro-level canalization', which is a network-level property.

Willadsen and Wiles have analyzed the macro-level canalization properties of the non-spatial or the single-cell subnetwork of the *Drosophila* segment polarity network (SPN) [136]. One of their main findings is that each of the five basins of attraction of the model converging on to wild-type attractors could be redescribed with very few schemata[5]. One of them, for example, is a set of

---

[5] Their method of schema-redescription is not the same as the one introduced by Marques-Pita and Rocha, in that

49152 configurations redescribed with just two schemata that is also the minimum required number of schemata to redescribe that set. The authors formulate ‘observed coherency’ and ‘structural coherency’ as measures of *robustness* that express the extent to which attractor basins could be redescribed, relative to a null model. Accordingly, they find that the basins of attraction of the SPN model are highly coherent and structured, and thus underlies its robustness. In contrast, they find that the yeast cell-cycle model is robust not due to high macro canalization per se but due to very large basins (that in turn results in high macro canalization). Gates and Rocha [39] have also found that the way canalization operates in these models determines how controllable they are.

Marques-Pita and Rocha analyze the single-cell SPN as well in [88] with their schema-redescription framework. There, they find that a total of only six macro-level schemata (dubbed *minimal configurations* (MC)) are necessary to redescribe all the configurations that converge to the five attractors, which stands in contrast to the 24 that Willadsen & Wiles found. One of the MCs, for example, implies that of the 15 nodes in the model it is sufficient to set the states of nodes SLP and *nhh* to 1 and 0 respectively and be assured that the network will converge to a certain attractor. The authors go on to analyze the full 4-cell parasegment model consisting of 60 nodes whose dynamical landscape is too large to enumerate (2^{60}), and identify the set of necessary and sufficient conditions that guarantee convergence to wild-type attractors.

The central conclusion of their analysis is that micro-level canalization engenders macro-level canalization, thus providing a scalable method to analyze the dynamics, robustness and controllability of large multivariate dynamical systems, such as those used in systems biology models of biochemical regulation.

---

their method does not consider overlapping schemata while the latter does.

Chapter 2

Effective connectivity and criticality

In this chapter, we test the hypothesis that the mean effective connectivity $\langle k_e \rangle$ of a Boolean network is a better predictor of its dynamical regime than its mean in-degree $\langle k \rangle$.

We approach this hypothesis as a binary classification problem: what set of Boolean network ensemble parameters best predict the observed dynamical regime, as measured by the Derrida coefficient (§ 1)? In particular, does a set of parameters utilizing $\langle k_e \rangle$ better predict the dynamical regime than one utilizing $\langle k \rangle$ in its place? If yes, how best to characterize the criticality transition in terms of $\langle k_e \rangle$?

We present the argument that motivates the hypothesis that mean effective connectivity improves the prediction of dynamical regime and characterization of criticality, followed by a description of the experimental setup and analysis of results confirming the hypothesis. The chapter concludes with an analysis of actual models of biochemical systems in the light of our new theory of criticality based on effective connectivity.

2.1 Motivation

The intuition behind the hypothesis is the following: while k specifies the number of inputs variables that an automaton is seemingly regulated by (apparent connectivity), k_e specifies the number of inputs variables that actually control it on average. Thus, effective connectivity is an estimate of the “truer” connectivity of a BN, which should be a better predictor of criticality than in-degree. The following analytical argument expands on this point.

Lower k_e means an automaton with larger input redundancy (k_r). That is, we observe fewer schemata with more wildcards after redescribing its LUT(§ 1.6.2)—the automaton is built of fewer “building blocks”. Therefore, the random flip of an input is less likely to flip the automaton’s state.

Let $x = f(i_1, \ldots, i_k)$ denote the automaton x and its Boolean function f of k inputs, and let $\langle a \rangle$ denote the expected probability that x negates its state when exactly one of its input states is negated (§ 1.6.5). In the absence of the schema-redescription of f, we consider every 2^k possible input states as a schema, and we can calculate a rough estimate of $\langle a \rangle \approx \langle a \rangle_k$ can be calculated as follows. For every schema that goes to a given output we calculate the probability of randomly choosing, with replacement, another schema that goes to the opposite output when a random input is perturbed:

$$
\begin{aligned}
\langle a \rangle &= \frac{1}{k} \sum_{j=1}^{k} a_j \qquad (2.1) \\
&= \frac{1}{k} \sum_{j=1}^{k} \Pr\big(f(i_1, \ldots, i_j, \ldots, i_k) = 1, f(i_1, \ldots, \tilde{i_j}, \ldots, i_k) = 0\big) \\
&\qquad\qquad + \Pr\big(f(i_1, \ldots, i_j, \ldots, i_k) = 0, f(i_1, \ldots, \tilde{i_j}, \ldots, i_k) = 1\big) \\
&= \frac{1}{k} \sum_{j=1}^{k} \Pr\big(x = 1\big) \cdot \Pr\big(f(i_1, \ldots, \tilde{i_j}, \ldots, i_k) = 0 \,|\, x = 1\big)
\end{aligned}
$$

$$
\begin{aligned}
&\quad + \Pr\big(x=0\big)\cdot \Pr\big(f(i_1,\ldots,\tilde{i_j},\ldots,i_k)=1 \,|\, x=0\big)\\
&= \frac{1}{k}\sum_{j=1}^{k}(p)\cdot \Pr\big(f(i_1,\ldots,\tilde{i_j},\ldots,i_k)=0 \,|\, x=1\big)\\
&\quad + (1-p)\cdot \Pr\big(f(i_1,\ldots,\tilde{i_j},\ldots,i_k)=1 \,|\, x=0\big)\\
&\approx \frac{1}{k}\cdot k\cdot p\cdot \frac{(1-p)2^k}{2^k-1} + \frac{1}{k}\cdot k\cdot(1-p)\cdot\frac{p2^k}{2^k-1}\\
&= \frac{2^k 2p(1-p)}{2^k-1}\\
&= \langle a\rangle_k
\end{aligned}
$$

The approximation step in the above assumes that the probability that toggling an input result in the flipping of the state of x is simply given by the probability (with replacement) of the occurrence of that state (the actual probability, without replacement, depends on p and k that can be accurately computed using a probability tree). This is tantamount to calculating the probability of randomly choosing a schema that goes to the opposite state, out of the $2^k - 1$ schemata (2^k minus the schema that x is currently in), as described above.

With the additional knowledge of the canalization of automaton x—its schema redescription—function f is revealed as comprised of fewer schemata. If we simplify the problem by assuming that all schemata have the same number of wildcards (same input redundancy for each building block of the LUT), then we have[1] a total of $2^{k_e} < 2^k$ schemata of which $p2^{k_e}$ schemata go to output 1 ($x = 1$), and $(1-p)2^{k_e}$ schemata go to output 0 ($x = 0$). Therefore, a more nuanced estimate of $\langle a\rangle \approx \langle a\rangle_{k_e}$ accounting for the presence of canalization can be calculated as follows. For every schema going to a given output we calculate the probability of randomly choosing, with replacement, another schema going to the opposite output when a random *effective* input is perturbed (perturbing a redundant input is guaranteed to not flip the output):

---

[1] In reality, typically, the schemata obtained overlap in their redscription of the automaton's LUT and have distinct number of wildcards as can be seen in the examples presented in § 1.

$$
\begin{aligned}
\langle a \rangle &= \frac{1}{k} \sum_{j=1}^{k} a_j \qquad (2.2) \\
&= \frac{1}{k} \sum_{j=1}^{k_e} \ \Pr\big(f(i_1, \ldots, i_j, \ldots, i_k) = 1, f(i_1, \ldots, \tilde{i_j}, \ldots, i_k) = 0\big) \\
&\qquad + \Pr\big(f(i_1, \ldots, i_j, \ldots, i_k) = 0, f(i_1, \ldots, \tilde{i_j}, \ldots, i_k) = 1\big) \\
&= \frac{1}{k} \sum_{j=1}^{k_e} \ \Pr\big(x = 1\big) \cdot \Pr\big(f(i_1, \ldots, \tilde{i_j}, \ldots, i_k) = 0 \,|\, x = 1\big) \\
&\qquad + \Pr\big(x = 0\big) \cdot \Pr\big(f(i_1, \ldots, \tilde{i_j}, \ldots, i_k) = 1 \,|\, x = 0\big) \\
&= \frac{1}{k} \sum_{j=1}^{k_e} \ (p) \cdot \Pr\big(f(i_1, \ldots, \tilde{i_j}, \ldots, i_k) = 0 \,|\, x = 1\big) \\
&\qquad + (1-p) \cdot \Pr\big(f(i_1, \ldots, \tilde{i_j}, \ldots, i_k) = 1 \,|\, x = 0\big) \\
&\approx \frac{1}{k} \cdot k_e{\cdot}p \cdot \frac{(1-p)2^k}{2^k - 2^{k-k_e}} + \frac{1}{k} \cdot k_e \cdot (1-p) \cdot \frac{p2^k}{2^{k-k_e}} \\
&= \frac{1}{k} \cdot k_e{\cdot}p \cdot \frac{(1-p)2^{k_e}}{2^{k_e} - 1} + \frac{1}{k} \cdot k_e \cdot (1-p) \cdot \frac{p2^{k_e}}{2^{k_e} - 1} \\
&= \qquad \frac{k_e}{k} \cdot \frac{2^{k_e} 2p(1-p)}{2^{k_e} - 1} \\
&= \qquad \langle a \rangle_{k_e}
\end{aligned}
$$

The approximation step is analogous to the one in Eqn. 2.1, except now the probability of flip of the state of x upon toggling an input is now determined by k_e rather than k: it is the probability of randomly choosing a schema that goes to the opposite state, out of the $2^{k_e} - 1$ schemata.

When $k_e = k$, Eq. 2.2 reduces to Eq. 2.1. Both Eqs. 2.1 and 2.2 suggest that $\langle a \rangle$ approaches $2p(1-p)$ as $k \to \infty$ and $k_e \to k \to \infty$, respectively; in other words, it is equivalent to a random guess biased by p. Conversely, for smaller values of k and k_e, the same equations suggest that $\langle a \rangle$ deviates further away from a random guess. More importantly, Eq. 2.2 suggests that, given a small k_e and a large k, $\langle a \rangle_{k_e}$ deviates further away from a random guess, whereas $\langle a \rangle_k$ approaches it. In summary, the smaller k_e/k is the larger is the discrepancy between the estimates of Eqs. 2.1 and

2.2, with $\langle a \rangle_{k_e}$ becoming the more accurate estimate of the probability of the automaton changing its state given a random input flip (Fig. 2.1).

For example, consider a canalizing automaton with only one canalizing input, that is $k_e = 1$, which is possible only when $p = 0.5$. For this BF, $\langle a \rangle_{k_e} = 1/k$ is exact—it approaches 0 for large k which is expected as most input bit flips are redundant since there is only one canalizing input. In contrast, $\langle a \rangle_k = 2^{k-1}/(2^k - 1)$ approaches 0.5 (the value of $2p(1-p)$) for large k which is a poor estimate of the real behavior of a highly canalizing automaton where most input perturbations are likely not to affect the state of the automaton.

The reasoning above shows how knowledge of effective connectivity yields a more accurate characterization of the dynamical response of individual Boolean automata to random input perturbations (bit-flips), than what we can obtain when we do not consider canalization. Therefore, we expect it to play a big role in the dynamical behavior—especially criticality—of networks of such automata, which we investigate next.

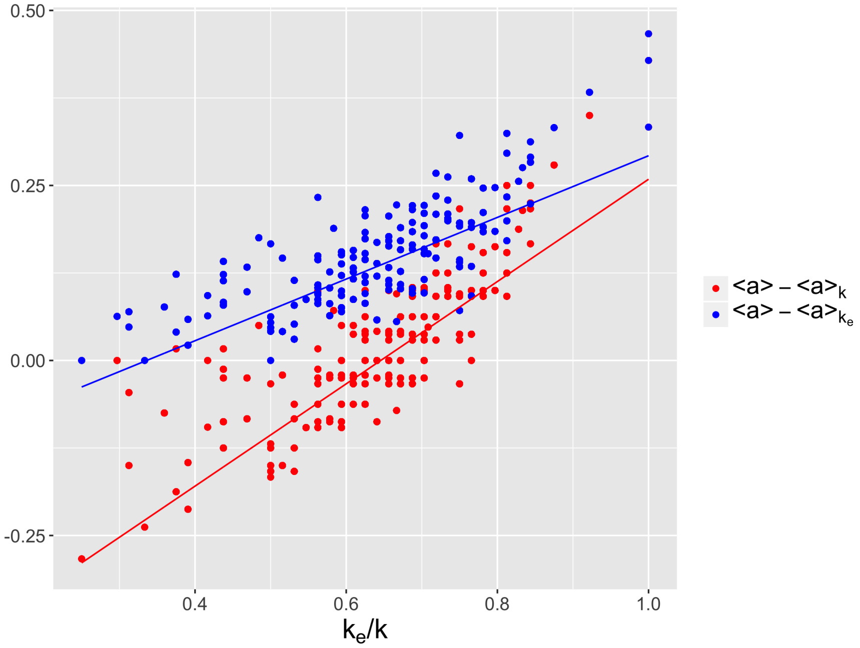


Figure 2.1: Difference between the true values of $\langle a \rangle$ and the estimates, namely $\langle a \rangle_k$ and $\langle a \rangle_{k_e}$, as a function of the ratio k_e/k computed for all possible automata of $k \in \{2,3,4\}$. The linear regression lines just serve as generic trends; they show that the discrepancy in the estimate provided by $\langle a \rangle_k$ (red line) becomes larger than that of $\langle a \rangle_{k_e}$ (blue line) as k_e/k becomes smaller.

2.2 Prediction of Dynamical Regime with Effective Connectivity

We follow a computational approach to compare k and $\langle k_e \rangle$ as predictors of criticality of RBNs:

1. Generate RBN ensembles parameterized by both (k, p) and (k_e, p) over an appropriate range of values
2. Compute their dynamic regimes based on D_c (§ 1.5.2)
3. Evaluate classification performance of statistical models with said parameters as predictors of dynamical regime.

2.2.1 RBN Ensemble Generation

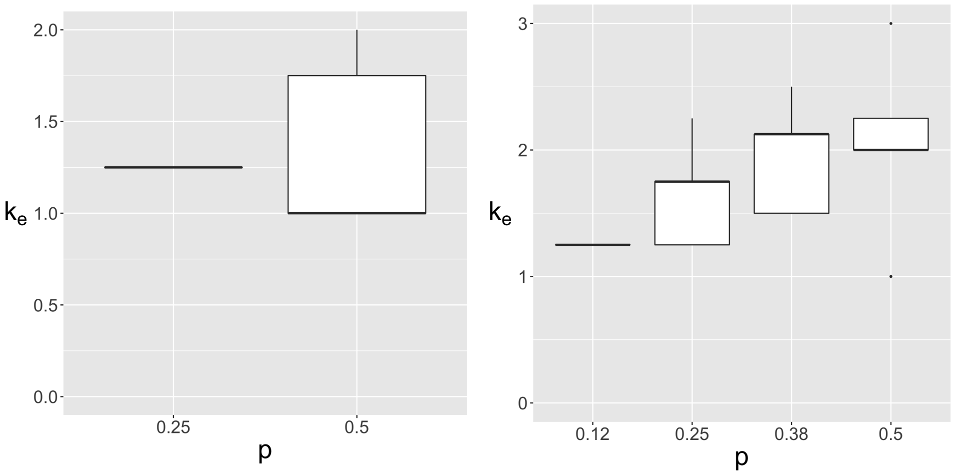
The experimental setup starts by generating RBN ensembles constrained by parameter set (k, k_e, p), so that each ensemble has both (k, p) and (k_e, p) parameters associated with it. An element of the (k, k_e, p) RBN ensemble is generated as follows: for each node choose a random set of k nodes as its inputs, then assign to it a random Boolean function with k inputs, output-bias p and an effective connectivity from a small range whose mean is equal to parameter k_e.

Producing a random Boolean function with a given (k, p) is straightforward: generate a random binary vector of length 2^k and assign it to the 2^k LUT entries listed in lexicographic order, which is the same as ordering by their decimal values. Simultaneously controlling for its k_e, however, is not as trivial. The solution we adopted is to generate a 'catalog' of Boolean functions with various combination of (k, k_e, p) values, from which Boolean with appropriate parameter values are picked during the generation of the RBN ensembles.

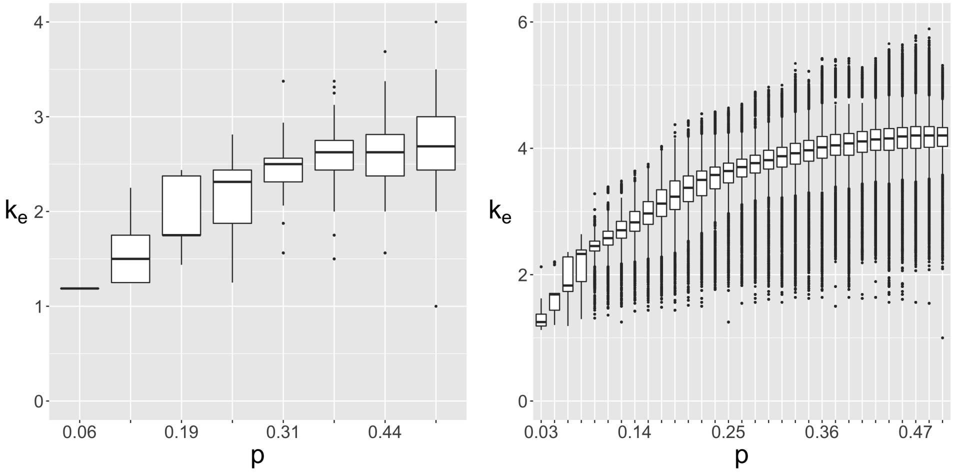
The catalogs for Boolean functions of $k = 2, 3, 4$ are exhaustive. For larger k, Boolean functions are first obtained by random generation for a given k and p, with their k_e subsequently computed. The number of possible Boolean functions for a given k and p is equal to $C(2^k, p2^k)$. Thus, for $k > 4$, the catalogs contain a random sample of 10^4 Boolean functions for each (k, p) if the total number possible is greater than 10^4. Additionally, to obtain Boolean functions with k_e in ranges rarely found by random generation via k and p alone, we employ a genetic algorithm (GA) to search the space of Boolean functions in such ranges of k_e as described below. A GA with binary tournament selection with mutation (rate set to 0.001) and no crossover, also known as 'microbial' GA [51], was used since it yielded the fastest optimization. The population in our case consists of LUT binary output vectors of length 2^k, and the fitness score of an individual is its k_e. An initial 'seed' population consisting of the BFs with the smallest possible k_e for the given k and p is constructed, which is accomplished by assigning 1 as the output to the first $p2^k$ ordered LUT entries and 0 to the rest. The GA steps are then iteratively performed until a sufficient number of Boolean functions within a given target range of k_e is obtained. Each range of k_e in the catalog computed via the GA contains at least 100 unique Boolean functions. Fig. 2.2 shows the distribution of k_e per p for the Boolean function catalogs for each k thus generated.

The following values of the ensemble parameters were considered: the number of nodes per network $N = 100$, $k \in \{2, 3, 4, 6, 8\}$, $p = (0, 0.5]$ with $\Delta p = 1/2^k$ whose lower bound is set at 0.01, and $k_e = [1, k]$ with $\Delta k_e \approx 0.5$. Importantly, although k and p is the same for all the nodes of a given BN in a random ensemble, the effective connectivity of each node is chosen from a small range of values, $[k_e - \Delta k_e/2, k_e + \Delta k_e/2]$. Since k_e, unlike k, varies smoothly (Fig. 1.7), there are very few Boolean functions with the same given value of k_e, therefore allowing functions in same BN to vary k_e within a small range allows us to maintain sufficient variety of Boolean functions and still parameterize the ensemble with a desired k, p, and k_e. By sweeping the ensemble parameters in this way, we generated a set of 100 networks for each parameter combination, resulting in the generation of a RBN set with a total of 266,400 samples, whose dynamical regimes were individually measured (§ 1.5.2).

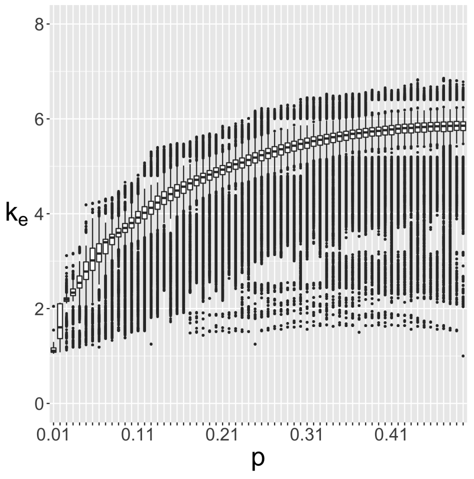
In the analysis that follows, we will refer to the individual BNs rather than ensembles. The parameters that identify an individual BN in an ensemble are the corresponding mean values $\langle k \rangle$, $\langle p \rangle$ and $\langle k_e \rangle$. As explained before, $\langle k \rangle = k$ and $\langle p \rangle = p$, that is, the k and p of all nodes in a BN are the same and equal to the associated ensemble parameter, while its $\langle k_e \rangle$ may be slightly different from the associated ensemble parameter k_e. In summary, we identify a BN by its k, p and $\langle k_e \rangle$.



(a) $k = 2$ **(b)** $k = 3$



(c) $k = 4$ **(d)** $k = 6$



(e) $k = 8$

Figure 2.2: Summary of the catalogs showing the distribution of k_e per p over a range of values of k. Catalogos for $k \in \{2, 3, 4\}$ are exhaustive; see main text for details.

2.2.2 Results of Experiments

We analyze the data obtained from the ensemble experiments described above to answer the following questions:

- How well do distinct statistical models based on BN parameters involving either k or $\langle k_e \rangle$ predict the observed dynamical regime?
- Are models utilizing $\langle k_e \rangle$ in place of k better predictors of dynamical regime?
- What is the best model of criticality that involves $\langle k_e \rangle$?

We formulate the above questions as a binary classification problem, and approach it in the following way. We evaluate the classification and regression performances of statistical models of varying complexities to select the best model, and then we estimate the generalization (classification and regression) performance of the best model via cross-validation.

First, we make a few important observations from the data. The dynamical regime of every RBN sample is shown in Fig. 2.3 separately in the k-p and $\langle k_e \rangle$-p spaces; Fig. 2.4 shows the same, except in aggregates for better visualization. Fig. 2.5 projects the dynamical regimes in k-$\langle k_e \rangle$ space. Note that only the ordered and chaotic regimes are displayed in all figures, as the critical regime is combined with the former (almost no RBNs with D_c exactly equal to 1 were found).

The first clear observation is that the dynamical regimes of RBNs are better separated in the $\langle k_e \rangle$-p space than in the k-p space. Indeed, we observe stable networks well into the area predicted to be chaotic by the existing criticality theory (eq. 1.6, § 1.5.3); we also observe chaotic networks in the space predicted to be stable for the same theory. It is particularly interesting that RBNs with $p = 0.5$ — which are most likely to be chaotic for any k according to (eq. 1.6, § 1.5.3 — with stable dynamics are observed at every k (Fig. 2.3a), but in the $\langle k_e \rangle$-p these networks cluster at a single value $\langle k_e \rangle = 1$ for every k (Fig. 2.3b). To complement these observations, Fig. 2.5 shows that $\langle k_e \rangle$ is a much more dominant discriminator of dynamical regimes than k, as indicated by the coefficients of the expression specifying the boundary separating the two regimes (known as the "criticality model" defined below). These observations immediately suggest the superior discriminating power of $\langle k_e \rangle$ over k, which we quantify below with models that include effective connectivity.

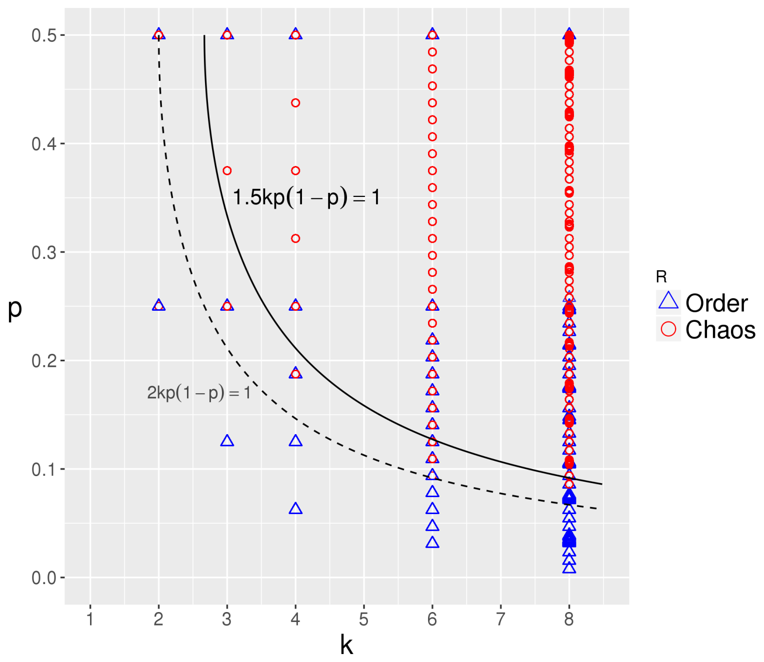
The boundaries separating the two regimes shown in Figs. 2.3 depict a multi-linear relationship among the respective predictors, as indicated by the respective expressions. This suggests that the actual relationship (since the one shown in the figures depict just one possibility) among the predictors may be expressed by a model in the space of multi-linear models (linear models with interactions) in both cases. Moreover, since the objective is to model dynamical regime as a binary variable R, and not the actual values of D_c. Specifically, we consider a binary decision mode where $R = 1$ iff $D_c > 1$ denoting Chaos, otherwise $R = 0$ denoting Order; a logistic transformation of candidate multi-linear models is required from which the binary "Class labels" (R) can further be inferred using a step transformation.

In every model, instead of considering p as a term, we use the term $p(1-p)$ due to the principle of duality of Boolean logic which states that there is no fundamental distinction between 0 and 1 in Boolean algebra. Thus, for every Boolean function f, there exists a dual f_d defined [28] as: $f_d(i_1, i_2, \ldots, i_k) = \neg f(\neg i_1, \neg i_2, \ldots, \neg i_k)$. That is, f_d is obtained by negating the 0s and 1s in both the condition (input) and transition (output) parts of the LUT entries[2] of f. For example, the dual of logical AND is logical OR. The duality principle further means that every BN $\mathcal{B}$ has a dual counterpart $\mathcal{B}_d$ whose automata nodes are the logical duals of the first, and which is dynamically indistinguishable, i.e., $D_c(\mathcal{B}) = D_c(\mathcal{B}_d)$. Therefore, the terms p and $(1-p)$ must be treated symmetrically in any model of criticality, since the characteristic output biases of $\mathcal{B}$ and its dual

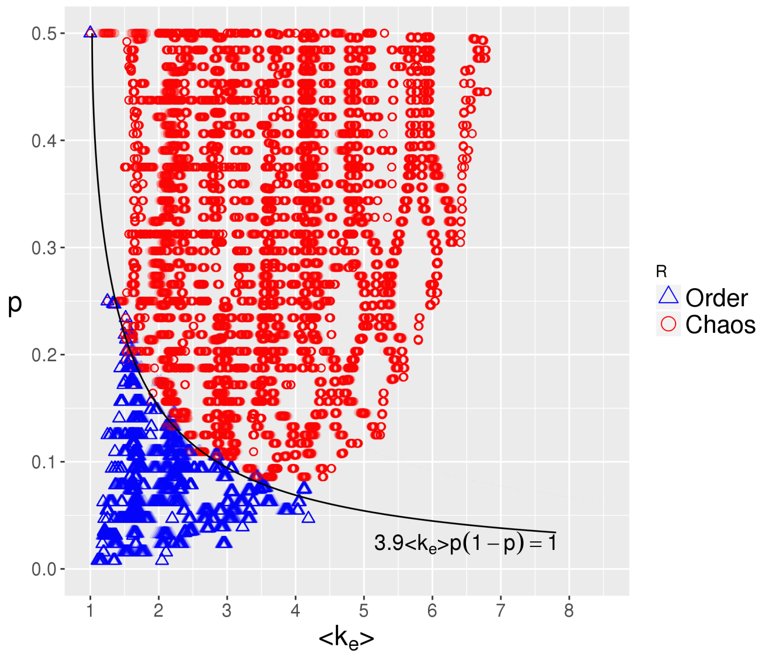
---

[2] The dual is thus not the same as the complement which is obtained by negating the 0s and 1s only in the outputs of the LUT entries

$\mathcal{B}_d$ are p and $1-p$ respectively. That is to say that the critical boundary must be symmetric about $p = 0.5$.

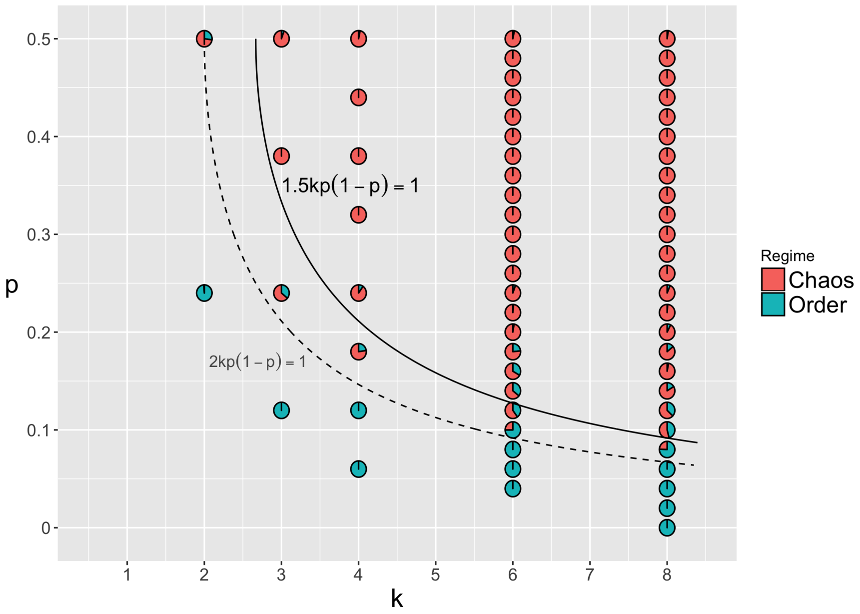


(a) *k*-*p* space.

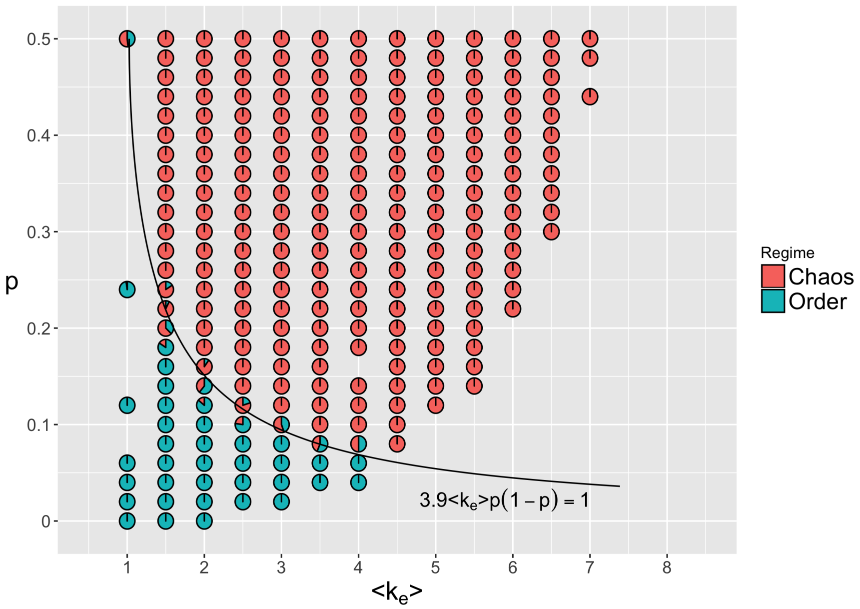


(b) $\langle k_e \rangle$-*p* space.

Figure 2.3: Phase diagrams in the *k*-*p* and $\langle k_e \rangle$-*p* spaces showing the dynamical regime of every RBN sample; blue triangles represent RBNs with ordered dynamics, and red circles chaotic dynamics. Critical boundaries displayed for Model class (2) (see main text). Also shown for reference in the *k*-*p* space is the critical boundary (eq. 1.6) as dictated by the existing theory of criticality.



(a) *k*-*p* space.



(b) $\langle k_e \rangle$-*p* space.

Figure 2.4: Phase diagrams in the *k*-*p* and $\langle k_e \rangle$-*p* spaces showing the dynamical regimes of aggregates of RBN samples; blue pie slices indicate the proportion of RBNs with ordered dynamics, and red indicates chaotic dynamics. Critical boundaries displayed for Model class (2) (see main text). Also shown for reference in the *k*-*p* space is the critical boundary (eq. 1.6) as dictated by the existing theory of criticality.

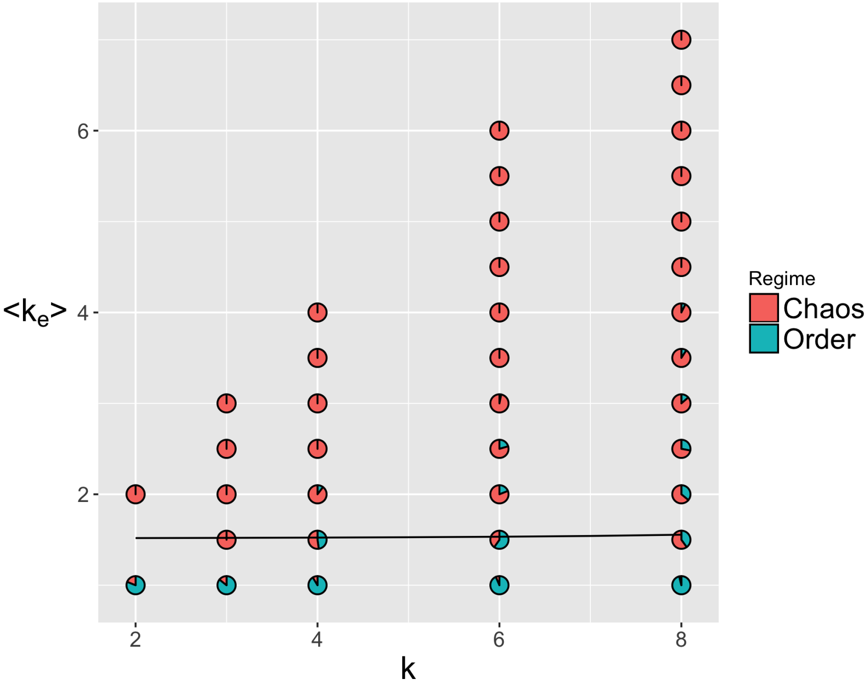


Figure 2.5: Phase diagram in the k-$\langle k_e \rangle$ space showing the dynamical regimes of aggregates of RBN samples; blue pie slices indicate the proportion of RBNs with ordered dynamics, and red indicates chaotic dynamics. The equation of the critical boundary displayed is: $0.1k + 0.7\langle k_e \rangle - 0.1k\langle k_e \rangle = 1$. The coefficient of the $\langle k_e \rangle$ term is 7 times larger than the coefficients of both the k term and the interaction term which are close to 0.

To compare the ability of effective connectivity to predict the dynamical regime of RBNs with that of in-degree, we consider a set of models with varying degrees of complexity. For each complexity level, we compare equivalent models for k and $\langle k_e \rangle$. In other words, for each class of models, we consider a version using mean connectivity k and a version using the mean effective connectivity $\langle k_e \rangle$ instead. This allows us to quantify how much the inclusion of information about canalization in our models improves prediction performance.

The general form of all models is:

$$R = \text{step}(\text{logistic}(\text{Model})) \tag{2.3}$$

where the output of the logistic function is the probability of observing Class label '1' (Chaos), and the output of the step function is the predicted Class labels: '1' (Chaos) if the output of the logistic function is greater than a *threshold* (T) of 0.5; and '0' (Order) otherwise. Thus, we have both a regression model and a binary classifier; the step function turns the logistic model into a classifier.

Each Model belongs to one of the following "Model classes", listed in increasing order of model complexity indicated by the number of terms and the number of predictors in each term:

1. **a)** $c_1.k$; **b)** $c_1.\langle k_e \rangle$
2. **a)** $c_1.k.p(1-p)$; **b)** $c_1.\langle k_e \rangle.p(1-p)$
3. **a)** $c_1.k + c_2.p(1-p)$; **b)** $c_1.\langle k_e \rangle + c_2.p(1-p)$
4. **a)** $c_1.k + c_2.k.p(1-p)$; **b)** $c_1.\langle k_e \rangle + c_2.\langle k_e \rangle.p(1-p)$
5. **a)** $c_1.k.p(1-p) + c_2.p(1-p)$; **b)** $c_1.\langle k_e \rangle.p(1-p) + c_2.p(1-p)$
6. **a)** $c_1.k + c_2.k.p(1-p) + c_3.p(1-p)$; **b)** $c_1.\langle k_e \rangle + c_2.\langle k_e \rangle.p(1-p) + c_3.p(1-p)$

Finally, we derive a *criticality model* from each fully specified (with regressed coefficients) model expression by setting it to 0 which translates to the probability of observing Class label '1' (or '0') being equal to 0.5. In other words, a criticality model specifies the critical boundary surface separating the Order and Chaos regimes. We further re-express each criticality model where the intercepts are eliminated and the right-hand side is equal to 1, so that the resulting expression is similar in form to the current theory of criticality (Eqn. 1.6).

For example, if the regressed Model is:

$$-2 + 3.k.p(1-p)$$

Then, the corresponding criticality model is:

$$-2 + 3.k.p(1-p) = 0$$

$$\implies 1.5.k.p(1-p) = 1$$

so that it predicts Class '1' if the value of $1.5.k.p(1-p)$ is greater than 1, and Class '0' otherwise. Henceforth, by "criticality model" we refer to the left-hand side of the corresponding expression, since the R.H.S is always assumed to be 1.

We proceed with the statistical analysis in two consecutive steps:

1. **Select the best characteristic model:** Here, we evaluate the in-sample performance of all models in order to select the model in each Model class that best characterizes the data at hand. That is, we measure the performance (using measures described below) of the models fit to the full dataset to evaluate how well they explain the dynamical regimes of the RBNs at hand. Then, based on the model complexity and the in-sample performance, we select a

single best model that is both simple and performs well. In other words, this step evaluates the fit of all models in order to select the best model.

2. **Estimate the generalization performance of the best characteristic model:** Here, we estimate the out-of-sample performance of the best model obtained in the previous step. That is, we use the same performance measures as above to estimate the ability of the best model to generalize to unseen RBNs, via a cross-validation procedure described below. In other words, this step evaluates the prediction ability of the best model.

The following performance measures were used to evaluate the models:

- McFadden's r-squared (R^2): A measure of "pseudo" R^2 or goodness-of-fit for logistic regression models, computed as 1 minus the ratio of the log-likelihood of the model to that of the intercept-only model [21]. The maximum value of R^2 is 1.
- Matthews correlation coefficient (MCC) defined as follows [131]:

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}}$$

where,

True positive (TP) = Number of Class labels observed as '1' and classified as '1'

False positive (FP) = Number of Class labels observed as '0' and classified as '1'

True negative (TN) = Number of Class labels observed as '0' and classified as '0'

False negative (FN) = Number of Class labels observed as '1' but classified as '0'

Positive (P) = Total number of observed Class labels '1' $= TP + FN$

Negative (N) = Total number of observed Class labels '0' $= TN + FP$

MCC ranges between -1 and 1, where -1 indicates perfect opposite classification, 1 indicates perfect classification and 0 indicates random classification.

- Area under curve (AUC) defined as follows [135] (implemented in R package `ROCR`):

  $$AUC = \int_{1}^{0} TPR(T)FPR'(T)dT$$

  where,

  True positive rate (TPR) = Proportion of true positives in the positive Class $= TP/P$

  False positive rate (FPR) = Proportion of false positives in the negative Class $= FP/N$

  and interpreted as the probability with which the classifier ranks positive instances (Class label '1') higher than negative instances (Class label '0')[3]. In other words, the expected TPR under a uniform distribution of the possible choices of FPR, as determined by the value of T [47]; $T = 0 \implies TPR = 1, FPR = 1$ and $T = 1 \implies TPR = 0, FPR = 0$. As an area measure, AUC ranges between 0 and 1.

Here, the positive class is associated with the chaotic dynamical regime $R = 1$, and the negative class with the ordered regime $R = 0$. We also use R^2 to evaluate the models, given the classification problem at hand, because it is a standard goodness-of-fit metric for logistic models, and also relates to the model comparison significance tests described below.

These measures provide a well-rounded quantification of classification performance. In particular, MCC is ideal to measure classification in unbalanced scenarios, where one class has many more instances than the other; here, about 84% of the nets are in the Chaos class [8]. Note that if the classifier groups all data items into only one class, then MCC becomes an indeterminate value; we assume MCC to be equal to 0 in such cases because the classifier is really unable to differentiate between the classes. For example, if all items are classified as positive, then the values of TN and FN are equal to 0, turning the denominator of the MCC calculation to 0.

---

[3] Thereby, indirectly determining the probability with which positive instances are placed above the threshold T and negative instances below.

We define the performance difference between Models in a given class as follows:

$$\text{Performance difference} = \frac{\text{Performance(Model x.b) - Performance(Model x.a)}}{\text{Performance(Model x.a)}} \tag{2.4}$$

where $x \in \{1, \ldots, 6\}$.

We also used statistical significance tests, namely 'Vuong test' and 'Clarke test', which are designed for comparing non-nested regression models (neither model's predictors are a subset of the other, which is the case for the model pairs in every Model class) [27, 114]. The null hypothesis is that the two models are equivalent, and therefore the alternative hypotheses is that one of the models is better than the other, depending on the value of the test statistic. In Vuong test, the test statistic is the difference between the sums of log-likelihoods of models (x.a) and (x.b), which follows a normal distribution under null hypothesis. In Clarke test, the test statistic is the total number of outcomes where the log-likelihood of (x.a) is larger than that of (x.b), which follows a binomial distribution under null hypothesis. Here, the likelihood values are provided by the outputs of the logistic regression models for each data point before the step transformation in Eqn.3.4. We used a significance level of p-value = 0.05 to reject the null hypothesis and pick one of the two alternative hypotheses: if the probability of observing a test statistic value equal to or larger than the observed value is less than the significance level, then we conclude that (x.a) is significantly better than (x.b); and, if the probability of observing a test statistic equal to or smaller than the observed value is less than the significance level, then we conclude that (x.b) is significantly better than (x.a).

The performance measures are depicted in Fig. 2.6, and the performance differences in Fig. 2.7. For reference, Table. 2.1 lists the full models obtained, along with coefficients and performance scores. It is clear in every Model class that the version based on $\langle k_e \rangle$ substantially outperforms the version based on k. In the simplest Model class (1), the version with k (no canalization considered) yields MCC=0 because it classified every RBN instance into the chaos class (the largest class). In contrast, the version with $\langle k_e \rangle$ performs much better with an MCC ≈ 0.5, a performance close to the current theory, Model (2.a), with an MCC ≈ 0.75. This means that even without considering the bias parameter, a model based solely on effective connectivity yields a decent performance, whereas connectivity does not classify dynamical regime at all (MCC=0).

To select the best Model class from among the full suite of 6 Model classes, we use a 'pareto front' that depicts models arranged in increasing order of complexity and performance where a model is marked if and only if its performance is greater than that of all models of lower complexity (Fig. 2.8). The pareto front shows that the second simplest class of models, namely Model class (2), yields a level of performance close to the performances of the more complex model classes, while performing significantly better than the less complex Model classes—for both the k and $\langle k_e \rangle$ versions. The MCC of Model (2.a) is 0.73, which outperforms that of (1.a) with MCC = 0, while it is outperformed by only about 4.1% by even the most complex Model class (6.a). Likewise, the MCC of Model (2.b) is 0.96, which outperforms that of (1.b) by about 96%, while it performs equally well as even the most complex Model class (6.b). Similar trends can be observed for R^2 and AUC in qualitative terms. In summary, with very little addition in complexity Model class (2) achieves a significant performance gain, while even a significant amount of additional complexity over Model class (2) yields only a relatively slim gain in performance. Thus, we conclude, per 'Ockham's razor', that Model class (2) is the more characteristic class of models that is optimal in terms of both simplicity and performance.

Within Model class (2), we observe that the performance of Model (2.b) is much better than that of (2.a): the MCC of the former is better than that of the latter by about 32%, and the R^2 by about 59% (Fig. 3.11). Moreover, (2.b) also a significantly better fit than (2.a) as indicated by both Vuong and Clarke tests (Table. 2.1). Thus, we conclude that $\langle k_e \rangle$ is a significantly better predictor of dynamical regime than k, and that Model (2.b) is likely close to the actual model of criticality.

To estimate the robustness of the coefficients and the performance scores, we performed repeated random stratified-sub-sampling in-validation. Our subsamples are randomly chosen subsets containing 60% of the original RBN dataset, where both the proportions of RBNs in the order versus chaos regimes was preserved for each k. A set of 100 sub-samples was thus generated for each Model class and used to fit both models in the class and their performances were measured. Table. 2.2 summarizes the results of in-validation. We observe that the mean performance, as well as the mean values of model coefficients obtained by fitting across sub-samples, hardly deviate from what was observed in the full RBN set. Moreover, the standard deviation is very small for all both the performance measures and model coefficients. This demonstrates that the main results presented in Table. 2.1 are reliable for the parameter ranges considered here.

Class	Criticality model	R^2	MCC	AUC	Vuong test	Clarke test
1.a	-0.09.k	0	0	0.53	NO	NO
1.b	$0.63.<k_e>$	0.32	0.49	0.88	YES	YES
2.a	1.49.k.p(1-p)	0.59	0.73	0.96	NO	NO
2.b	$3.93.<k_e>.p(1-p)$	0.94	0.96	1	YES	YES
3.a	-0.21.k + 3.37.k.p(1-p)	0.63	0.75	0.96	NO	NO
3.b	$-0.03.<k_e> + 4.16.<k_e>.p(1-p)$	0.95	0.96	1	YES	YES
4.a	0.07.k + 4.86.p(1-p)	0.64	0.74	0.96	NO	NO
4.b	$0.21.<k_e> + 4.07.p(1-p)$	0.84	0.88	0.99	YES	YES
5.a	3.63.p(1-p) + 0.83.k.p(1-p)	0.64	0.75	0.97	NO	NO
5.b	$0.24.p(1-p) + 3.79.<k_e>.p(1-p)$	0.95	0.96	1	YES	YES
6.a	0.03.k + 3.99.p(1-p) + 0.53.k.p(1-p)	0.64	0.76	0.97	NO	NO
6.b	$-0.09.<k_e> + -0.82.p(1-p) + 5.12.<k_e>.p(1-p)$	0.95	0.96	1	YES	YES

Table 2.1: List of criticality models with coefficients and performance scores. Each grey-white banded pairs of rows are the model-pairs to be compared. The grey models utilize parameter k, while the white models utilize $\langle k_e \rangle$; all other parameters assume the same role in both models. The last two columns indicate whether a model was deemed as significantly better than the other in the Model class. If neither model is better than the other (null hypothesis is not rejected), then a 'NO' would be listed for both models. Models are arranged in increasing order of complexity from top to bottom. The best characteristic Model class (2) is highlighted in red.

Class	Criticality model	Coefficients	R^2	MCC	AUC	Vuong test	Clarke test
1.a	c1.k	-0.09±0	0±0	0±0	0.53±0	0%	0%
1.b	c1.$<k_e>$	0.63±0	0.32±0.004	0.49±0.005	0.88±0	100%	100%
2.a	c1.k.p(1-p)	1.49±0.005	0.59±0.002	0.73±0	0.96±0.002	0%	0%
2.b	c1.$<k_e>$.p(1-p)	3.93±0.003	0.94±0.004	0.96±0.005	1±0	100%	100%
3.a	c1.k + c2.k.p(1-p)	-0.21±0.003, 3.36±0.025	0.63±0.002	0.75±0.001	0.96±0	0%	0%
3.b	c1.$<k_e>$ + c2.$<k_e>$.p(1-p)	-0.03±0, 4.16±0.007	0.95±0	0.96±0	1±0	100%	100%
4.a	c1.k + c2.p(1-p)	0.07±0, 4.86±0.011	0.64±0.005	0.74±0.004	0.96±0.004	0%	0%
4.b	c1.$<k_e>$ + c2.p(1-p)	0.21±0.003, 4.07±0.009	0.84±0.001	0.89±0.005	0.99±0	100%	100%
5.a	c1.p(1-p) + c2.k.p(1-p)	3.63±0.023, 0.83±0.005	0.64±0	0.75±0.001	0.97±0.005	0%	0%
5.b	c1.p(1-p) + c2.$<k_e>$.p(1-p)	0.24±0.011, 3.79±0.008	0.95±0.002	0.96±0	1±0	100%	100%
6.a	c1.k + c2.p(1-p) + c3.k.p(1-p)	0.03±0.004, 3.99±0.032, 0.53±0.018	0.64±0.001	0.75±0.007	0.97±0.005	0%	0%
6.b	c1.$<k_e>$ + c2.p(1-p) + c3.$<k_e>$.p(1-p)	-0.09±0.004, -0.82±0.034, 5.12±0.04	0.95±0	0.96±0	1±0	100%	100%

Table 2.2: List of criticality models with the central tendencies and standard deviations of coefficients and performance scores obtained from repeated random stratified-sub-sampling in-validation. Each grey-white banded pairs of rows are the model-pairs to be compared. The grey models utilize parameter k, while the white models utilize $\langle k_e \rangle$; all other parameters assume the same role in both models. The last two columns indicate the proportion of comparison tests where a model was deemed as significantly better than the other in the Model class. Models are arranged in increasing order of complexity from top to bottom. The best characteristic Model class (2) is highlighted in red.

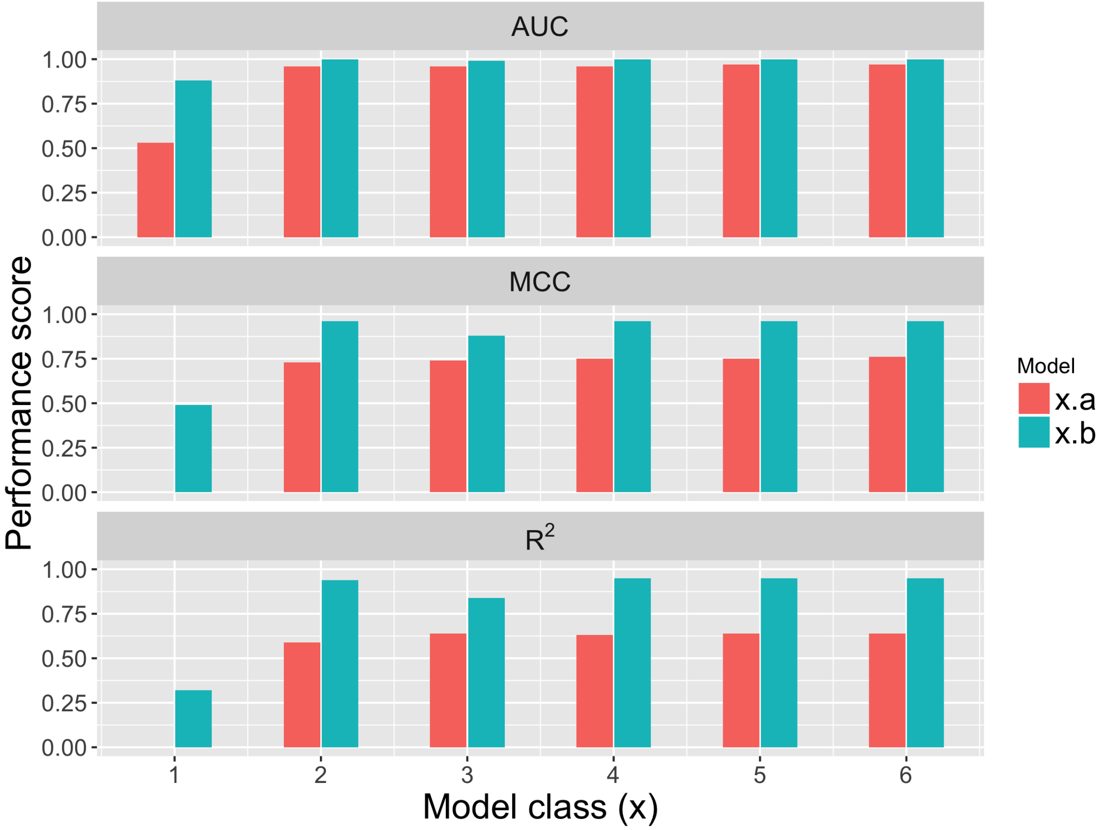


Figure 2.6: Performance of various regression models on the full set of RBNs. Models are arranged in increasing order of complexity from left to right. Because performance is computed on full set, results are absolute and not statistical.

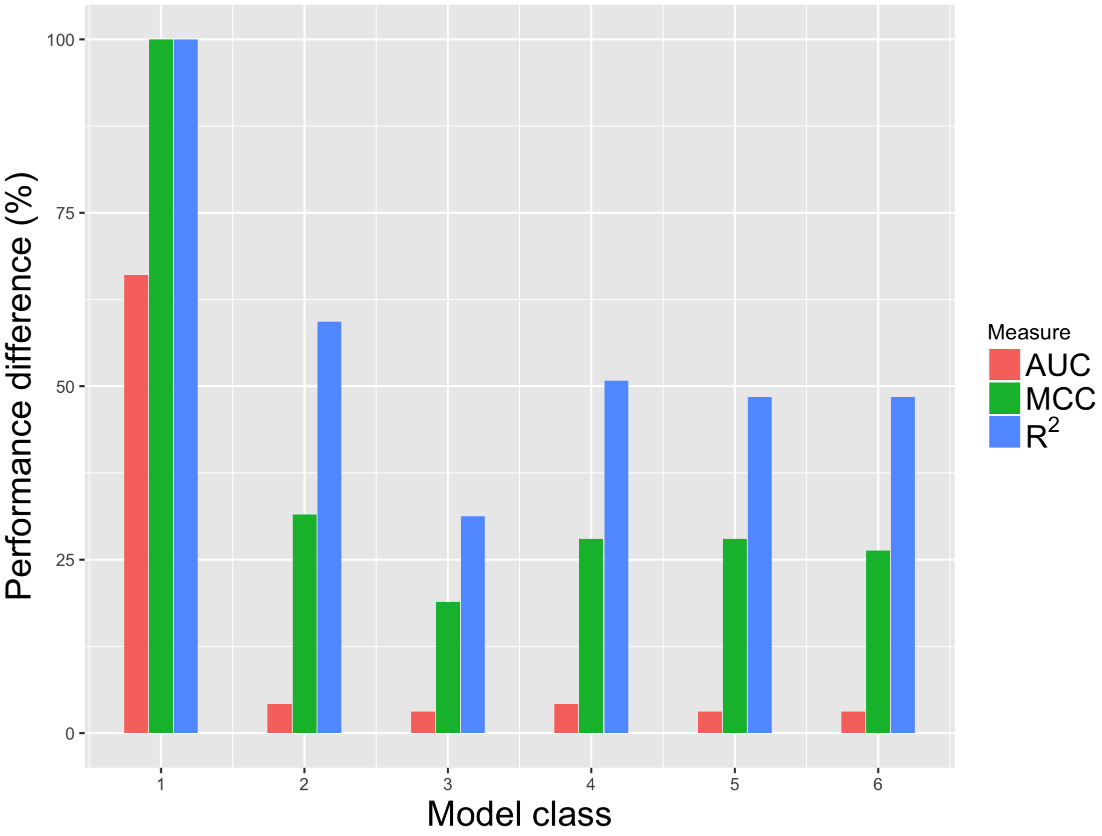


Figure 2.7: Performance differences of various regression models on the full set of RBNs. Models are arranged in increasing order of complexity from left to right. Values that shoot to ∞ due to a 0-denominator are capped at 100%.

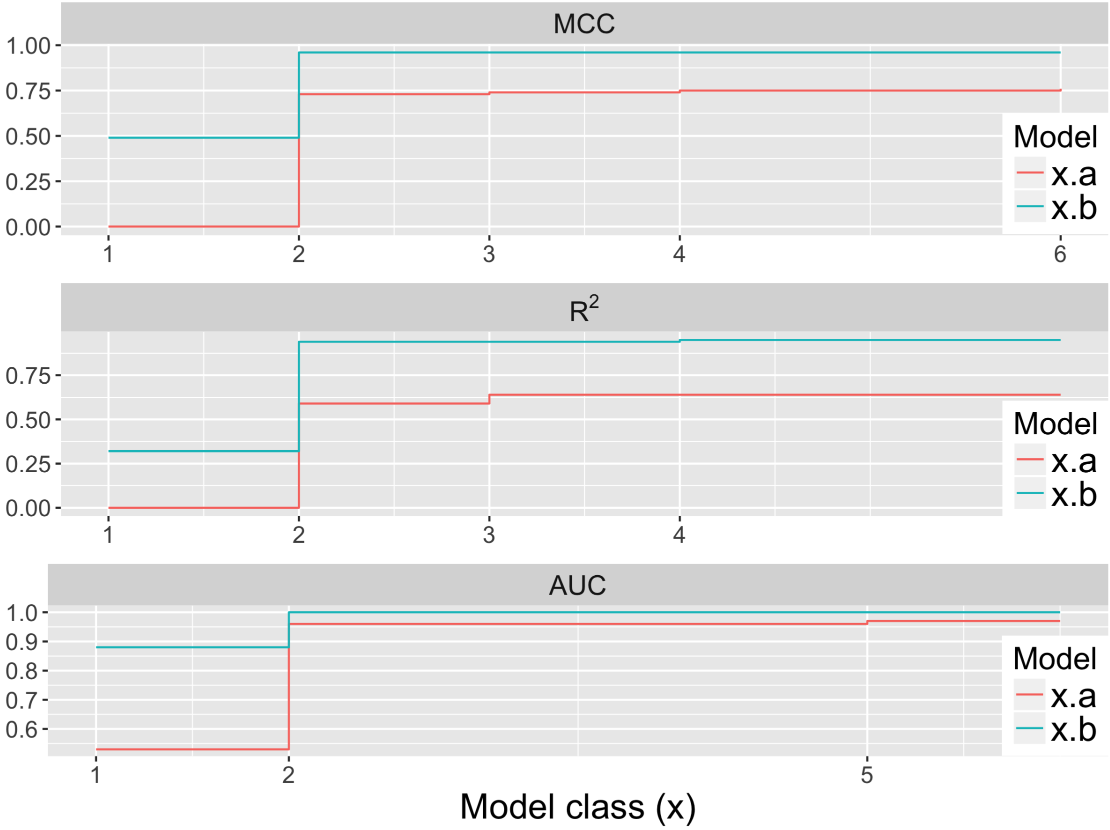


Figure 2.8: A 'pareto front' of the model complexity versus performance for the full suite of 6 model classes fit to the full dataset. Models are arranged in increasing order of complexity from left to right. A model class is marked (labeled on the axis) if and only if its performance is greater than the performances of all models of lower complexity.

To make sure the results above do not derive from overfitting the data, and to estimate generalization performance on out-of-sample RBNs, we performed cross-validation on the best Model class (2). More specifically, we performed nested 4-fold cross-validation by using modules implemented in packages `penalized` [44] and `CMA` [119]:

1. The full RBN dataset was randomly split into 4 non-overlapping equally sized partitions (75%-25% splits). This was repeated 4 times, thus yielding 4 *outer foldings*, and *both* Models (2.a) and (2.b) were evaluated on the same foldings.

2. In each outer folding, each of the four 25% splits was treated as the 'testing split' and the other 75% split as the 'training split', yielding a total of 16 outer training-testing pairs. A penalized logistic regression model with an 'L2' regularization parameter that penalizes the L2-norm of the model coefficients [76] was fit to the training split and evaluated on the testing split. The optimal value of the L2 parameter was chosen as follows:

    2.1. The training split was itself randomly split into 4 non-overlapping equally sized partitions (75%-25% splits of the outer 75% folding). This was repeated 4 times, yielding a total of 16 *inner* training-testing pairs. A penalized logistic regression model was fit to each inner training split over a range of L2 parameters. Mean performance on the inner testing splits was measured using log-likelihood. The L2 parameter that yielded the maximum cross-validated log-likelihood was chosen to train the outer training split.

3. Measures of classification and regression performance (as with the full dataset) on the testing splits were collected. The 16 sets of performance scores were averaged to produce an estimate of generalization performance score for each measure.

The results of cross-validation are summarized in Fig. 2.9 and Table. 2.3. We observe that all three cross-validated performance scores of Model (2.b) are significantly better than that of (2.a), and (2.b) is also a significantly better generalizable model than (2.a) per the model comparison significance tests. Thus, we conclude that not only Model (2.b) is a better characteristic model than (2.a), it also generalizes significantly better than (2.a). Moreover, the cross-validated performance scores of Model (2.b) are essentially the same as the ordinary performance scores, and can therefore be used as a generic model to make predictions of dynamical regimes of homogeneous RBNs in general for the RBN parameter ranges considered here.

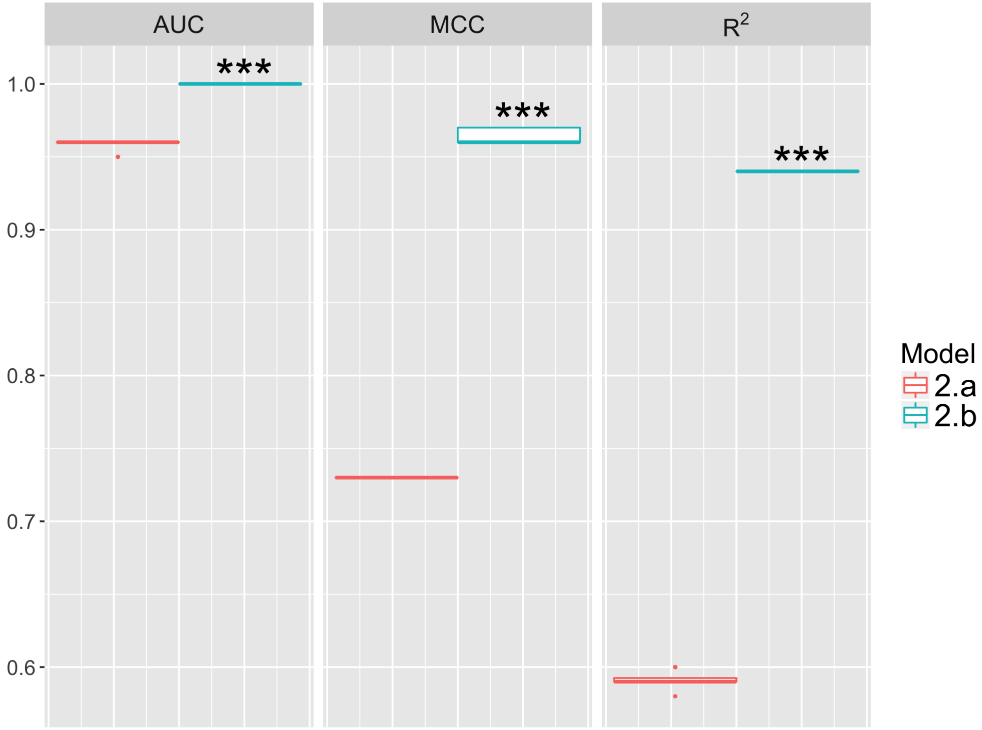
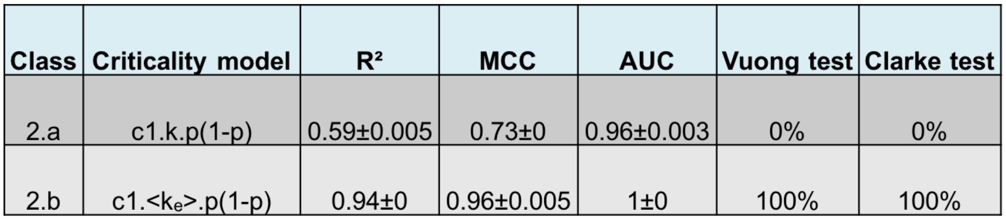


Figure 2.9: Generalization performance of Model class (3) from nested 4-fold cross-validation. The significance codes above the boxes indicate p-values ($< 0.001 =$ '***') obtained from a one-sided paired-sample t-test where variances of the samples were not assumed to be the same. Here, paired-sample was assumed because the two models were evaluated on the same set of 16 test folds; and one-sided test means that the alternative hypothesis is that the mean score of Model (2.b) is greater than that of (2.a).

Finally, we compare our new best Model (2.b) based on effective connectivity with the current theory based on apparent connectivity which belongs to Model class (2.a) but with a slightly different coefficient than the one derived from our data: $2.k.p(1-p) = 1$. Fig. 2.10 and Table. 2.4



Class	Criticality model	R^2	MCC	AUC	Vuong test	Clarke test
2.a	c1.k.p(1-p)	0.59±0.005	0.73±0	0.96±0.003	0%	0%
2.b	$c1.<k_e>.p(1-p)$	0.94±0	0.96±0.005	1±0	100%	100%

Table 2.3: Central tendencies and standard deviations of the generalization performance of Model class (2) from nested 4-fold cross-validation. The last two columns indicate the proportion (out of 16) of comparison tests where a model was deemed as significantly better than the other in the Model class.

summarize the results. As expected, the new model is better than the current model of criticality on every count.

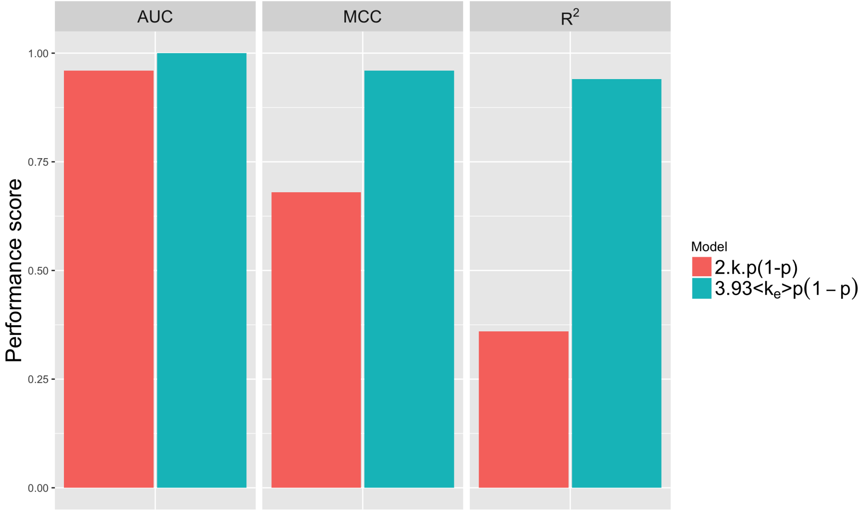
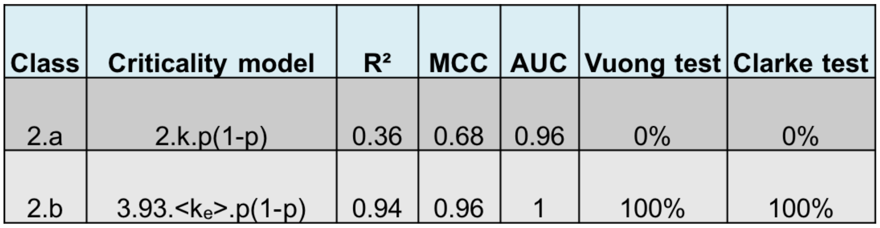


Figure 2.10: Performance of current model of criticality based on apparent connectivity versus new Model (2.b) based on effective connectivity on the full dataset.

We conclude that the new model of criticality based on effective connectivity, $3.93.\langle k_e \rangle.p(1-p) = 1$, is a revision to the current best model of criticality based on apparent connectivity, $2.k.p(1-p) = 1$, for homogeneous RBNs.



Class	Criticality model	R²	MCC	AUC	Vuong test	Clarke test
2.a	2.k.p(1-p)	0.36	0.68	0.96	0%	0%
2.b	3.93.<k_e>.p(1-p)	0.94	0.96	1	100%	100%

Table 2.4: Performance of current model of criticality based on apparent connectivity versus new Model (2.b) based on effective connectivity on the full dataset.

2.3 Effective Connectivity and criticality in Systems Biology Models

'The Cell Collective' was introduced as an online platform to create, store and analyze Boolean network models of biochemical pathways in cells [53]. Presently, the repository "`https://www.cellcollective.org/`" consists of 50 published models [52], examples of which are the lac operon interaction [128], T-cell receptor signaling [112], budding yeast cell cycle [58], apoptosis [84], cholestrol regulation [69], glucose repression [26], Influenza A replication [83], Drosophila body segmentation, lymphocytes differentiation [90], survival signalling in leukemic cells [141], and cardiac [54] and cortical area development [41].

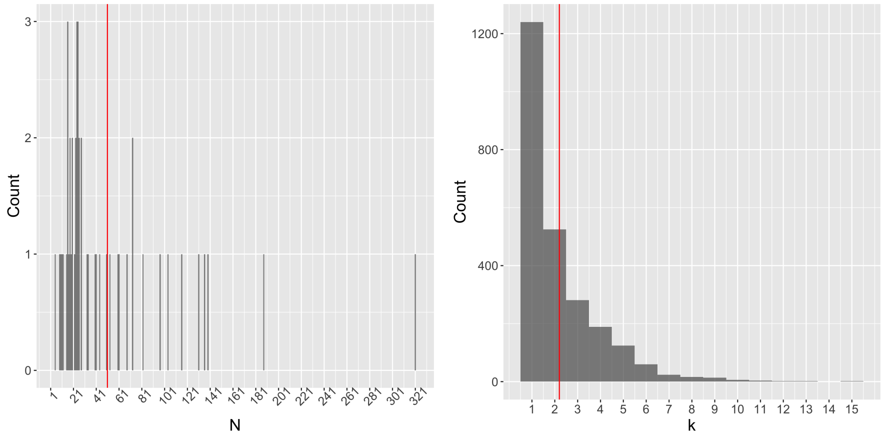
We analyzed a set of 49 of the 50 Boolean network models[4] found in this repository. Each model BN is comprised of N automata nodes, each with a distinct in-degree k, output bias p and effective connectivity k_e, all of which are summarized in Fig. 2.11.

It is clear from this figure that there is substantial canalization in the models as the mean $k_e \approx 1.18$ is substantially lower than mean $k \approx 2.2$, indicating that about 46% of the inputs are redundant or about 54% of the inputs are effective, on average. Moreover, since automata with $k = 1$ cannot be canalized (excepting tautology and contradiction), the degree of canalization is even greater

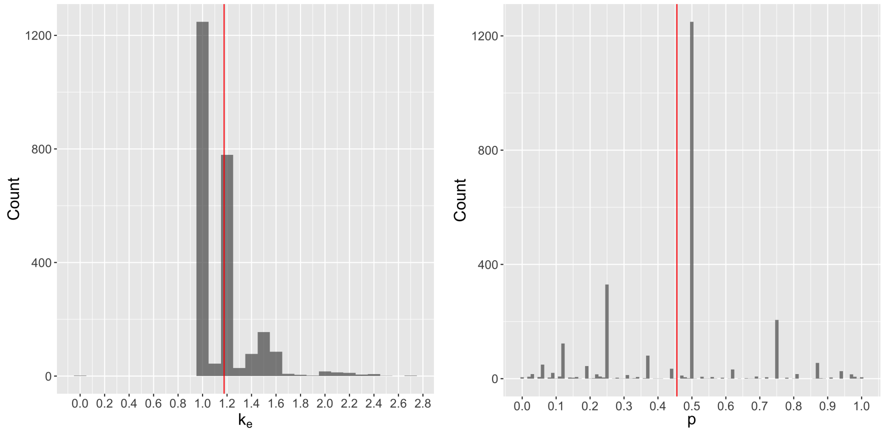
---

[4] The one that was left out had a node with so large an in-degree that an accurate computation of its k_e was not feasible. Although an estimate of k_e for this BN can be computed using faster heuristics-based methods like ESPRESSO [28], we left it out to ensure that all k_e were computed in the same manner.

for automata with $k > 1$: with the mean $k_e \approx 1.35$ and mean $k \approx 3.4$, about 60% of inputs per automaton are redundant. Indeed, as Fig. 2.12 shows, even automata with large k possess much lower effective connectivity—automata with $k > 10$ have a mean $k_e \approx 1.6$—indicating very large canalization in biological models.



(a) Histogram of N. **(b)** Histogram of k.



(c) Histogram of k_e. **(d)** Histogram of p.

Figure 2.11: (a) Histogram of the number of automata in the 49 BN models of the *Cell Collective*. (b-d) Histograms of in-degree (k), effective connectivity (k_e) and output bias (p) of all automata in the BN models of *Cell Collective*. Most of the nodes with $p = 0.5$ have $k = 1$, and most nodes with $p = 0.25$ and $p = 0.75$ have $k = 2$. The total number of automata considered is 2489. The vertical red line in each figure represents the mean value.

Fig. 2.13 depicts the Derrida coefficient D_c (§ 1.5.2) of the models. As can be seen, the mean D_c

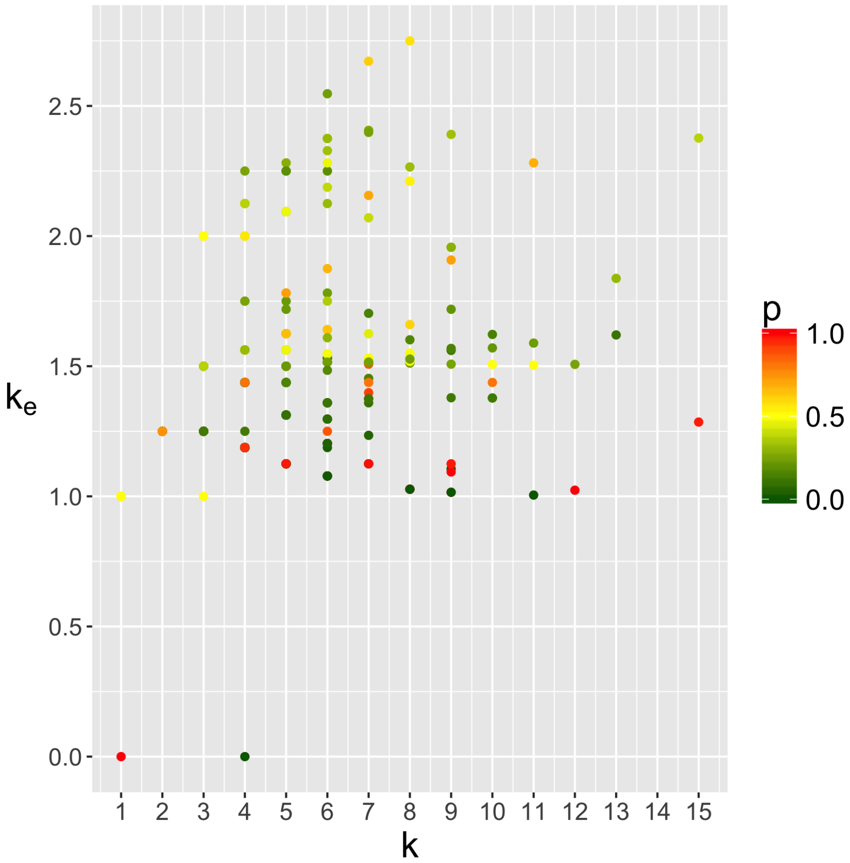


Figure 2.12: A map of k, k_e and p of the individual automata of the *Cell Collective* BNs.

is about 0.95, with most values clustering around it; about 80% of the D_c values lie within one standard deviation (≈ 0.07) of the mean. This suggests that most of these biochemical models display ordered dynamics (only 6 are slightly chaotic) and are close to being critical. This makes sense, since the Systems biology models in the *Cell Collective* tend to converge quickly to attractors representing cell types or other stable phenotypic behaviors (so ordered behavior is expected), but they also allow for adaptation allowing them to switch attractors under certain environmental conditions, which means that the networks cannot be too stable so that some perturbations could be allowed to propagate.

Since many of these models are almost critical, they should lie close to the critical boundary. Naturally, the parameter space where this occurs can be deemed to be the most characteristic of

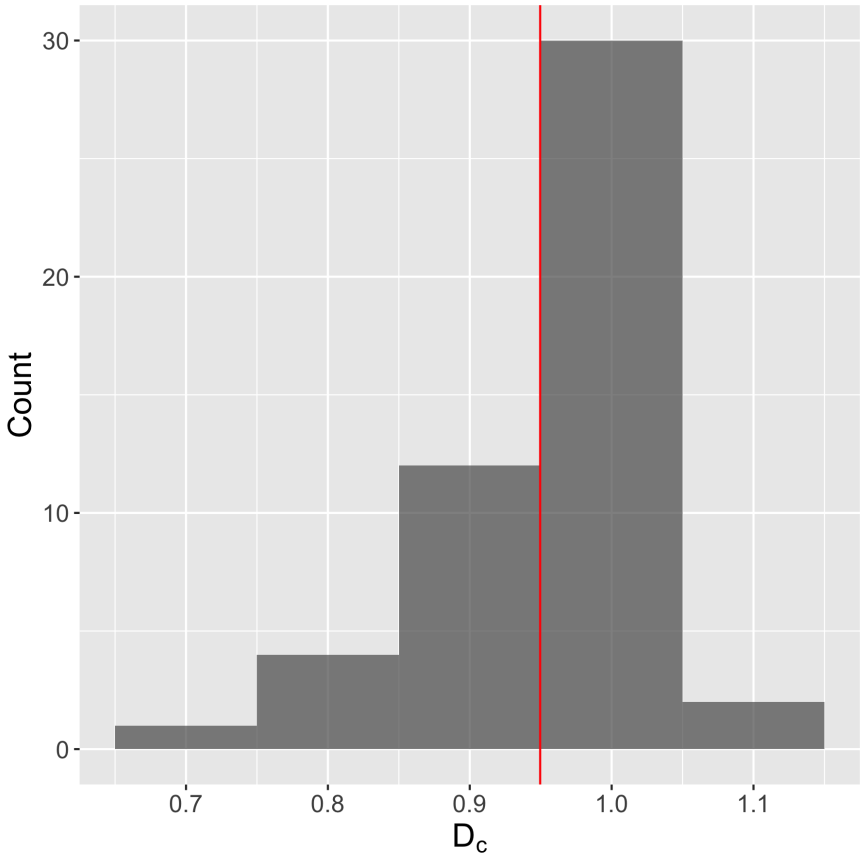
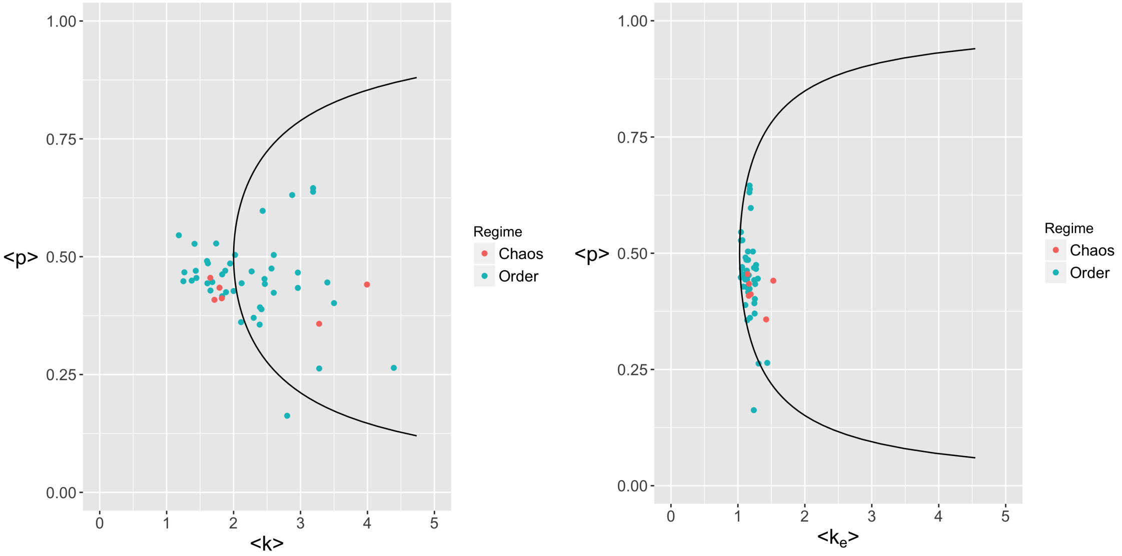


Figure 2.13: Distribution of D_c of the *Cell Collective* BNs. Red line represents the mean D_c.

these models in regards to criticality. To evaluate this hypothesis, we project the dynamical regimes of the models in two different parameter spaces, as show in Fig. 2.14: (a) $\langle k \rangle$-$\langle p \rangle$ and (b) $\langle k_e \rangle$-$\langle p \rangle$ spaces. The criticality models shown are respectively the current theory and the new theory based on k_e derived empirically in § 2.2.2. Here, $\langle k \rangle$, $\langle k_e \rangle$ and $\langle p \rangle$ represent the corresponding network-level mean values. In this sense, these figures clearly show that $\langle k_e \rangle$ is more characteristic than $\langle k \rangle$ of the models, as they tightly cluster around the critical boundary only in the case of $\langle k_e \rangle$ (Fig. 2.14b). This provides further evidence that a theory of criticality based on canalization is more characteristic of the dynamics of BN models of biochemical regulation.

The performance measures of both models are displayed in Table. 2.5. We observe that the $\langle k_e \rangle$ model is better than the k model only according to the AUC measure. Thus, the canalization-based classifier better ranks the dynamic regimes of the BNs. As per MCC and R^2, however, the k-based model performs better. These performance comparisons thus suggest that it is not possible to derive an unequivocal conclusion about whether $\langle k \rangle$ or $\langle k_e \rangle$ is more predictive of the dynamical regimes of the *Cell Collective* BNs.



(a) $\langle k \rangle$-$\langle p \rangle$ space.

(b) $\langle k_e \rangle$-$\langle p \rangle$ space.

Figure 2.14: Projection of the dynamic regimes of the *Cell Collective* BNs in the phase spaces derived by fitting models to homogeneous RBNs. The parameters $\langle k \rangle$, $\langle k_e \rangle$ and $\langle p \rangle$ represent the corresponding network-level means of the BNs. Critical boundaries shown here are those represented by the current theory (a), and obtained empirically (b) in Sec. 2.2.2.

Note however that we have used criticality models derived from homogeneous RBNs (§ 2.2) for the above analysis, whereas the *Cell Collective* BNs display higher heterogeneity particularly in k and p (Fig. 2.15). Therefore, we fit statistical models to only the *Cell Collective* dataset with the goal of deducing whether $\langle k \rangle$ or $\langle k_e \rangle$ is more *explanatory*, as opposed to being predictive, of the observed dynamical regimes.

The new phase spaces thus obtained are depicted in Fig. 2.16. From the performance scores, shown in Table. 2.6, we can conclude that $\langle k_e \rangle$ is indeed better than $\langle k \rangle$ at explaining the observed

Criticality model	R^2	MCC	AUC
$2.\langle k\rangle.\langle p\rangle(1-\langle p\rangle)$	-0.46	0.12	0.52
$3.93.\langle k_e\rangle.\langle p\rangle(1-\langle p\rangle)$	-7.35	0.08	0.77

Table 2.5: Performance of current theory based on $\langle k\rangle$ and new theory based on $\langle k_e\rangle$, derived in in Sec. 2.2.2, as evaluated on the *Cell Collective* dataset.

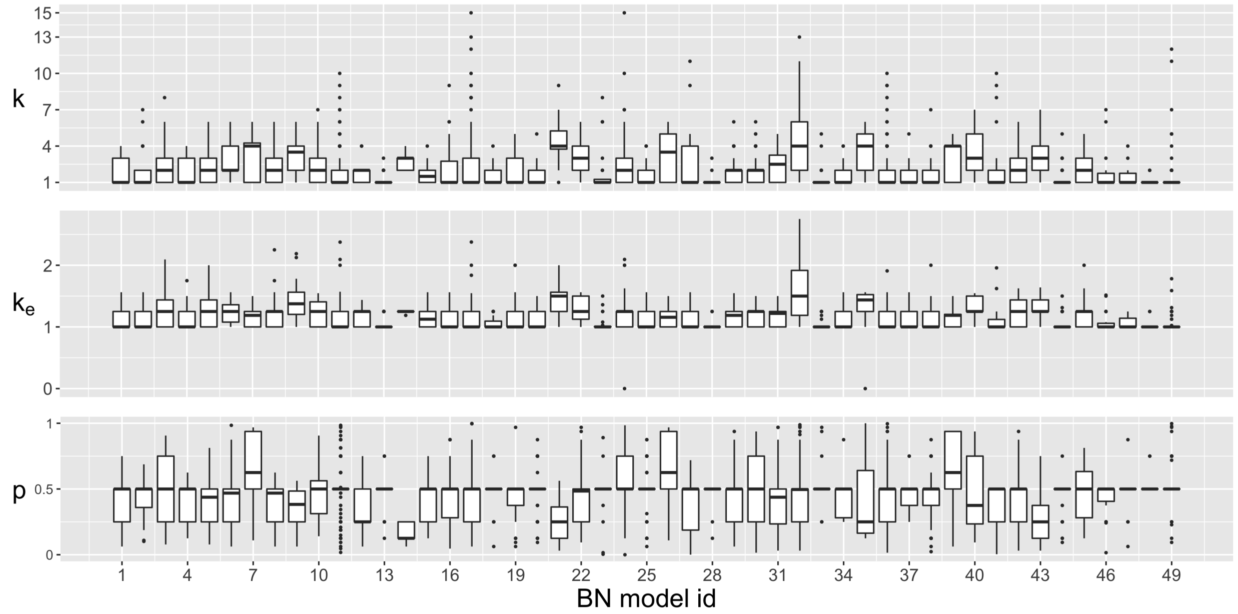
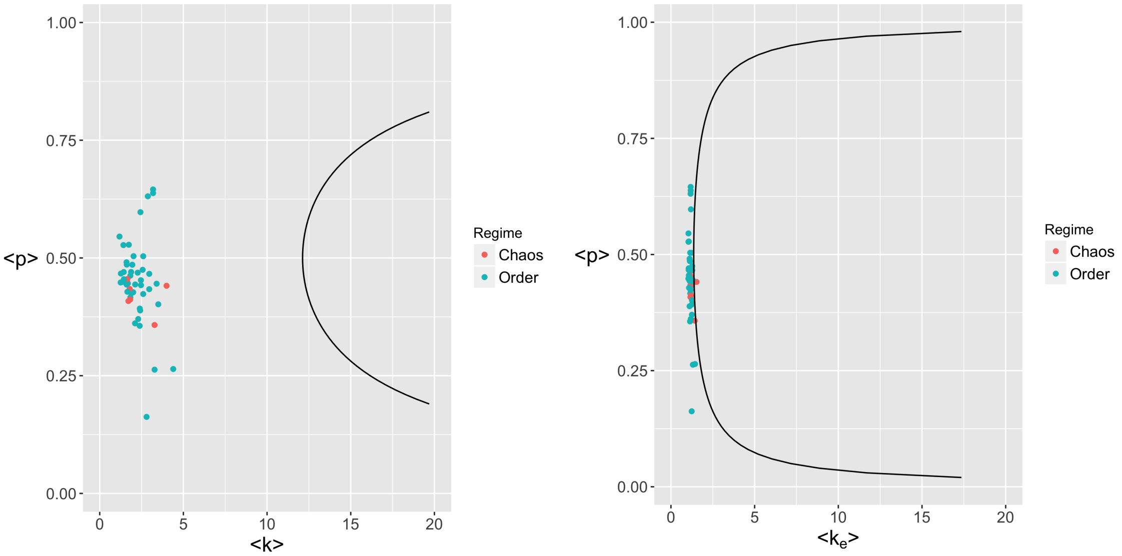


Figure 2.15: Distributions of k, k_e and p of the automata per BN model in *Cell Collective*.

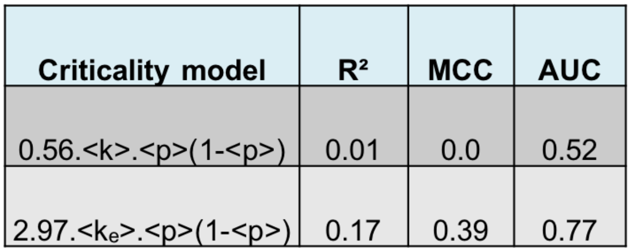
dynamical regimes of the *Cell Collective* BNs. In other words, the derived explanatory models suggest that canalization remains characteristic of dynamical regime also in the more realistic case of heterogeneous networks. In future work, we will verify this observation rigorously by considering RBNs with heterogeneous connectivity.



(a) $\langle k \rangle$-$\langle p \rangle$ space.

(b) $\langle k_e \rangle$-$\langle p \rangle$ space.

Figure 2.16: Projection of the dynamic regimes of the *Cell Collective* BNs in the phase spaces derived by fitting models to the *Cell Collective* data. The parameters $\langle k \rangle$, $\langle k_e \rangle$ and $\langle p \rangle$ represent the corresponding network-level means of the BNs. Critical boundaries shown here are those obtained empirically by fitting statistical models to the *Cell Collective* BNs.



Criticality model	R²	MCC	AUC
0.56.<k>.<p>(1-<p>)	0.01	0.0	0.52
2.97.<k$_e$>.<p>(1-<p>)	0.17	0.39	0.77

Table 2.6: Performance of the models based on $\langle k \rangle$ and $\langle k_e \rangle$ derived from regression and evaluation on the *Cell Collective* dataset.

Chapter 3

Unified canalization and criticality

In this chapter, we introduce the concept of *unified canalization* (k_u) as a measure of partial canalization that combines input redundancy and input symmetry. We further test the hypothesis that such a combined measure of canalization is a better predictor of criticality than just effective connectivity (§ 2). Our experiments show that the best statistical model utilizing k_u outperforms the counterpart model utilizing k_e in predicting dynamic regime by about 1% in terms of the MCC score. The performance difference further reveals a pattern upon conditioning the RBNs on k: up until $k = 8$, the k_e model outperforms the k_u model, but the difference consistently shrinks to the point where the k_u model outperforms the k_e model by about 2.2% at $k = 10$ and by about 4.7% at $k = 12$. Our results thus indicate that k_u is likely to become an even better predictor than k_e if more RBNs with larger values of k are considered. We conclude the chapter with an analysis of actual models of biochemical systems in the light of our new theory of criticality based on k_u.

3.1 Unified canalization

Unified canalization (k_u) is formulated to provide a deeper and a more nuanced measure of the amount of partial canalization in a Boolean automaton than input redundancy (k_r) or the complementary measure of effective connectivity (k_e). As a unified measure of effective connectivity and input symmetry, k_u provides a more accurate characterization of partial canalization, than k_e, of a Boolean automaton.

The unification of effective connectivity and input symmetry hinges on the fact that a wildcard schema is a type of two-symbol schema that has no permuting symbols [88]. As described in Eqn. 1.15, two-symbol schemata can be characterized by the following *schema parameters*: (1) n_l, the number of literals; (2) $n_f^{\#}$, the number of fixed (non-permuting) wildcards; and (3) n_g, the number of permuting symbols of which $n_g^{\#}$ are wildcards, n_g^1 are 1s and n_g^0 are 0s; where $n_l + n_f^{\#} + n_g = k$ and $n_g^{\#} + n_g^1 + n_g^0 = n_g$.

Wildcard schemata are characterized by the parameter values $n_g = 0$ and $n_f^{\#} \geq 0$ (e.g., $(1, 0, \#, \#)$), while the more general two-symbol schemata by $n_g \geq 0$ and $n_f^{\#} \geq 0$ (e.g., $(1, \#, \dot{\#}, \dot{1}, \dot{0})$).

We compute the set of all two-symbol schemata, F'', directly from F, just as done[1] for F'. A two-symbol schema, f''_θ, redescribes a subset of LUT entries: $\Theta_\theta \equiv \{f_\alpha : f_\alpha \rightarrowtail f''_\theta\} \subseteq F$. The set of all two-symbol schemata that redescribes the entire LUT is denoted as: $F'' \equiv \{f''_\theta\}$.

Our goal is to capture maximal canalization present in an automaton using the two-symbol schema redescription approach, hence to compute a F'' that satisfies *both* of the following properties:

1. F'' contains *maximal* two-symbol schemata. A maximal two-symbol schema is one such that there exists no other two-symbol schema in F'' that can redescribe the subset of LUT entries that it redescribes. That is, $(\Theta_\theta \nsubseteq \Theta_\phi) \wedge (\Theta_\phi \nsubseteq \Theta_\theta), \forall f''_\theta, f''_\phi \in F''$.
2. F'' is *complete*. It contains all possible all possible two-symbol schemata (satisfying some parameter combination) that redescribes the LUT. That is, $\nexists f''_\theta : \exists \Theta_\theta \equiv \{f_\alpha : f_\alpha \rightarrowtail f''_\theta\} \subseteq$

---

[1] Alternatively, F'' could in principle be computed from F'. However, since two-symbol schemata are just a more general type of schemata than wildcard schemata, we chose to compute them simply by expanding the procedure that computes wildcard schemata to look for two-symbol schemat as well.

$F \wedge f''_\theta \notin F''$.

The set F'' that satisfies the above properties may not be the same as the F'' computed as described in § 1.6.2; we refer to these sets as 'revised' and 'current' F'' respectively. In particular, the current F'' may not be complete in the sense described above. This is because certain two-symbol schemata cannot be obtained using the method by which the current F'' is computed—which is by combining wildcard schemata in F' containing the same number of wildcards, 1s and 0s [88]. The revised F'' captures these additional two-symbol schemata as well[2]; such schemata were first described in an earlier work [89], an example of which is illustrated in Fig. 3.1. That is, the revised F'' contains not only maximal two-symbol schemata but is also complete. As a consequence, it includes all wildcard and two-symbol schemata as the current F'', with an exception (explained below). As a consequence of the defining characteristics of the revised F'' stated above, it is possible that the set contains two-symbol schemata that may not be necessary to describe the automaton but are necessary to capture input symmetry. Furthermore, it is possible that the wildcard schemata that are deemed necessary in F' are rendered non-essential in F''. These are detailed further with the help of examples in Appendix. C. Henceforth, all references to F'' refer to the 'revised' set.

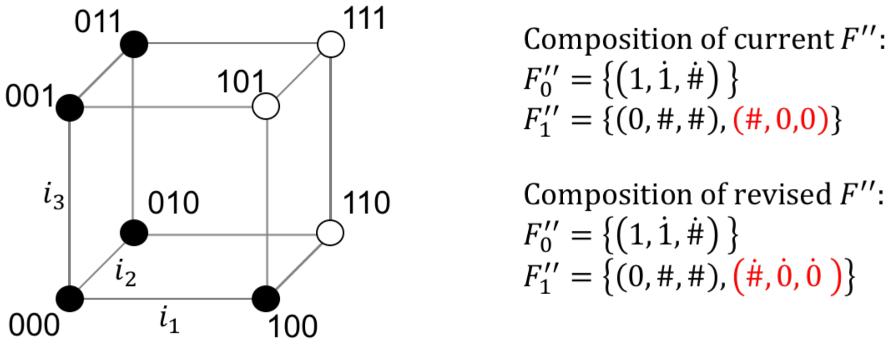
The algorithm to compute F'' for a Boolean automaton with k inputs proceeds by first identifying all two-symbol schemata that occur with either $n_f^\# > 0$ and $n_g = 0$, or those with $n_f^\# = 0$ and $n_g > 0$ in all possible 'subcubes' (§ 1.6.3) of a given size, and then combining subsets of these schemata in parallel-facing subcubes to produce other schemata that occur with $n_f^\# > 0$ and $n_g > 0$. The algorithm repeats these steps starting from the largest subcube (the full hypercube) and sequentially working its way through smaller ones[3]. In other words, the algorithm essentially adopts a brute-force approach to obtaining all possible two-symbol schemata[4]. For more details on

---

[2] Whether these new two-symbol schemata are significantly more informative when present than not, insofar as prediction of dynamical regime is concerned, is a question we save for furture work.

[3] The time complexity of this algorithm grows faster than 2^k but slower than $k!$.

[4] Hence, the revised F'' cannot miss two-symbol schemata obtained from other methods including the one used to



Composition of current F'':
$F_0'' = \{(1, \dot{1}, \dot{\#})\}$
$F_1'' = \{(0, \#, \#), (\#, 0, 0)\}$

Composition of revised F'':
$F_0'' = \{(1, \dot{1}, \dot{\#})\}$
$F_1'' = \{(0, \#, \#), (\dot{\#}, \dot{0}, \dot{0})\}$

Figure 3.1: A crucial difference between the compositions of the current and revised F''. Highlighted in red is an example of a schema contained in the latter (making it complete) but not in the former. The two-symbol schema $(\dot{\#}, \dot{0}, \dot{0})$ in the revised F'' is not present in the current F'' because it cannot be computed by combining the wildcard schemata $(0, \#, \#)$ and $(\#, 0, 0)$ since they don't contain the same number of wildcards, 1s and 0s.

the algorithm and for a worked example, see Appendix. A.

Although F'' is in principle complete, the version used in this dissertation misses certain types of two-symbol schemata, for reasons described below. Specifically, the algorithm detects only those two-symbol schemata with parameters $n_g^{\#} > 0$ when $n_g > 0$; it misses those with parameters $n_g^{\#} = 0$. Examples of schemata not detected by the algorithm are $(\dot{0}, \dot{1})$ (found in the redescription of logical XOR), $(\#, \dot{0}, \dot{1}, \dot{1})$ etc.

To assess the potential impact of the missing schemata on the criticality experiments below, we estimate the proportion of the missed schemata for various values of k. For this purpose, we enumerate schema 'signatures' and calculate the proportion of the signatures of the missed schemata. A schema signature is simply a particular combination of the schema parameter values. All schemata that satisfy a given set of parameter values are considered to have the same signature. For example, the schemata $(0, 0, \dot{1}, \dot{0}, \dot{\#})$ and $(\dot{0}, 1, \dot{1}, \dot{\#}, 1)$ have the same signature, even though they are different

---

compute the current F''

instances. Moreover, a signature is parity-symmetric: there is no difference between the permuting 1s and 0s. For example, the schemata $(\dot{0}, \dot{1}, \dot{1})$ and $(\dot{1}, \dot{0}, \dot{0})$ bear the same signature.

Schema signatures can be classified into three types, each satisfying a unique set of parameter constraints: (i) $n_f^\# \geq 0$ and $n_g = 0$; (ii) $n_f^\# \geq 0$, $n_g > 0$ and $n_g^\# > 0$; and (iii) $n_f^\# \geq 0$, $n_g > 0$ and $n_g^\# = 0$. The number of schema signatures in each type, denoted respectively by s_1, s_2 and s_3, is calculated as:

$$s_1 = \sum_{n_f^\#=0}^{k} 1 = 1 + k \tag{3.1}$$

$$s_2 = \sum_{n_f^\#=0}^{k-2} \sum_{n_g=2}^{k-n_f^\#} \sum_{n_g^\#=1}^{n_g-1} \sum_{n_g^1=0}^{\left\lfloor \frac{n_g-n_g^\#}{2} \right\rfloor} 1$$

$$s_3 = \sum_{n_f^\#=0}^{k-2} \sum_{n_g=2}^{k-n_f^\#} \sum_{n_g^1=1}^{\left\lfloor \frac{n_g}{2} \right\rfloor} 1$$

The above expressions hold for $k \geq 2$, since permutation needs at least two variables; for $k = 1$, therefore, $s_2 = s_3 = 0$. The expression for s_2 sweeps through all possible values of $n_f^\#$ from 0 up to a maximum of $k - 2$ (leaving at least two symbols for permutation), and for each value of $n_f^\#$, n_g is allowed to vary from 2 up to a maximum of $k - n_f^\#$, and for each value of n_g, $n_g^\#$ varies from 1 to a maximum of $n_g - 1$ (the case with all permuting symbols being wildcards is not allowed since such a schema is just a wildcard schema), and finally for each value of $n_g^\#$, n_g^1 is swept from 0 to a maximum that respects parity symmetry. The expression for s_3 is the same, except $n_g^\#$ stays at a fixed value of 0.

Fig. 3.2 illustrates the ratio $s_3^* = \frac{s_3}{s_1+s_2+s_3}$ as a function of k (green line), providing a rough estimate of the probability of observing the missing schemata in a random Boolean automaton with k inputs—the actual probabilities depend on various constraints like p, presence or absence of other schemata etc[5]. Even though there is an initial rise in s_3^* up to $k = 4$ where it attains the peak value of about 21%, it approaches 0 as $k \to \infty$, suggesting that the missing schemata may not play a crucial role in predicting criticality at large k, as they would not be expected to significantly contribute to the canalization of the automata[6].

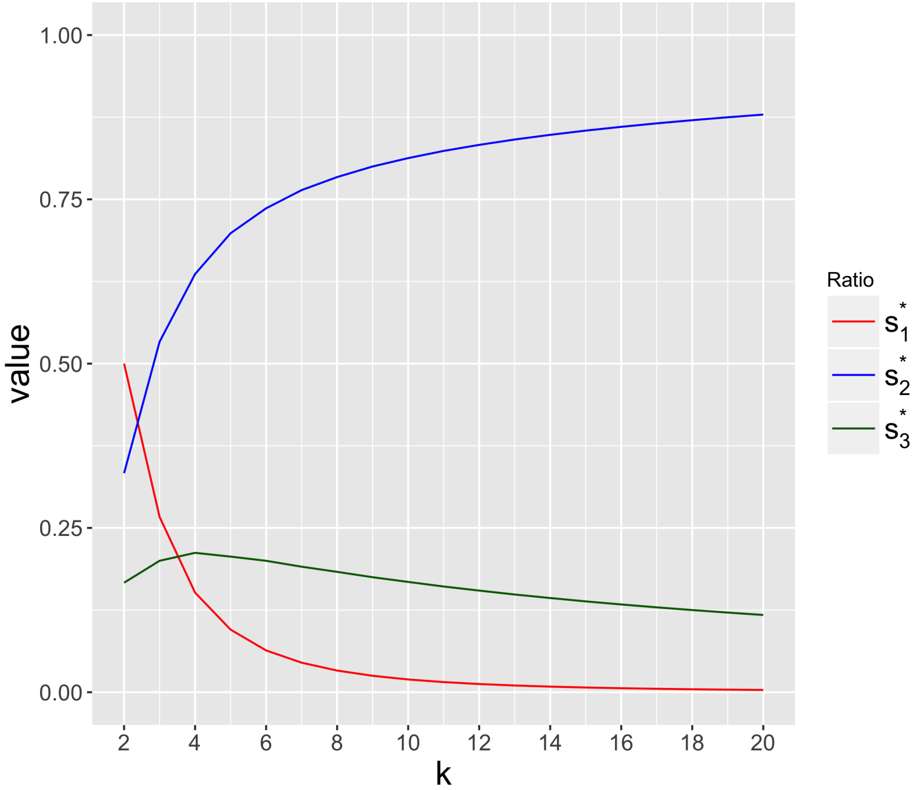


Figure 3.2: The ratios s_1^*, s_2^* and s_3^* as a function of k. The total number of schema signatures, $s_1 + s_2 + s_3$ rises with k: 5 for $k = 2$, 471 for $k = 10$, and 5366 for $k = 20$.

The amount of canalization contributed by a wildcard schema to a Boolean automaton is simply equal to $n_f^{\#}$ (§ 1.6.3). Here, we introduce a more general measure of the amount of canalization contributed by a generic two-symbol schema, namely its *dimension*.

---

[5] One way is to calculate it is to draw up a probability tree whose nodes are the schemata, and the probability of occurrence of a schema at a given level in the tree would depend on the schemata at every upper level of the corresponding branch and the total output bias p.

[6] However, in this dissertation we consider RBNs with only up to $k = 12$ where s_3^* varies between 15% and 21%, suggesting a possible but unlikely role of the missing two-symbol schemata in predicting criticality.

The dimension, $d_{f''}$, of a two-symbol schema f'' is tallied by the base-2-logarithm of its 'volume' $v_{f''}$—the cardinality of the corresponding set of redescribed LUT entries:

$$\begin{aligned} v_{f''} &= 2^{n_f^{\#}} \cdot \sum_{n=n_g^1}^{n_g^1+n_g^{\#}} \binom{n_g}{n} \\ d_{f''} &= \log_2(v_{f''}) \end{aligned} \tag{3.2}$$

which obeys the property $0 \leq d_{f''} \leq k$ since $1 \leq v_{f''} \leq 2^k$. The dimension of a wildcard schema is $n_f^{\#}$, since $d_{f''} = \log_2(2^{n_f^{\#}}) = n_f^{\#}$, as $n_g = n_g^{\#} = n_g^1 = 0$. Two-symbol schemata have fractional dimensions in general, with the exception of wildcard schemata whose dimensions are always whole numbers, as just shown. For example, the dimension of $(1, \dot{1}, \dot{\#})$ is about 1.58, while that of $(1, \#, \#)$ is 2. Note that dimension, as defined here, is not a unique descriptor of two-symbol schemata. For example, both the schemata $(1, \#, \#)$ and $(\dot{1}, \dot{1}, \dot{\#})$ have the same dimension equal to 2; although, in this case the conserved dimension may be indicative of the fact the first schema has fewer variables that are fully redundant compared to the second which has more variables that are less redundant.

Furthermore, the dimension of a schema can be expressed as a proportion of the total number of fixed wildcards and permuting symbols—the symbols that contribute to canalization—$n_s = n_f^{\#} + n_g$, as follows:

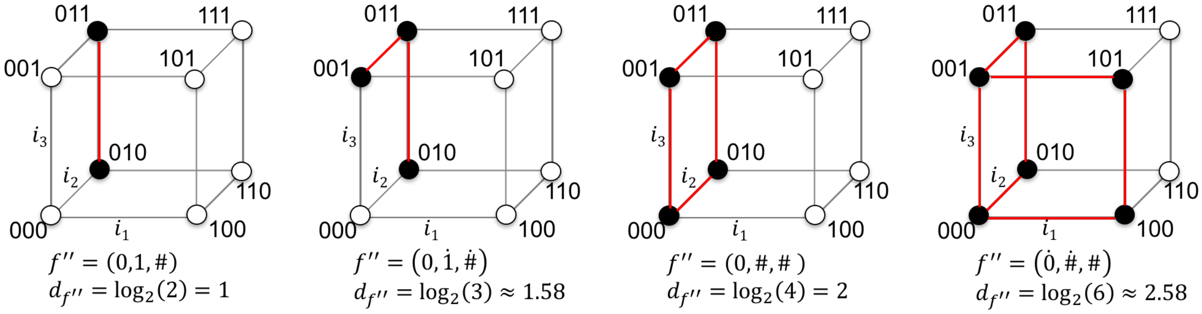
$$\begin{aligned} d_{f''} &= \log_2(v_{f''}) \\ &= n_s \times \frac{\log_2(v_{f''})}{\log_2(2^{n_s})} \\ &= n_s \times \log_{2^{n_s}}(v_{f''}) \end{aligned}$$

where, $0 \leq \log_{2^{n_s}}(v_{f''}) \leq 1$ is a normalization term that attains maximal value when all n_s symbols are fixed wildcards. This expression of $d_{f''}$ especially clarifies its character as a general measure of canalization by which the particular types namely redundancy and permutation or their combination can be expressed.

The dimension of a schema can also be understood as its geometrical dimension when visualized in a hypercube, as illustrated in Fig. 3.3. In particular, it is analogous to the ‘box-counting’ dimension of a fractal S which is defined as [134]:

$$\dim(\mathrm{S}) = \lim_{\epsilon \to 0} \frac{\log N(\epsilon)}{\log(1/\epsilon)}$$

where, $N(\epsilon)$ is analogous to $v_{f''}$. The difference is that measuring the dimension of a schema does not require the infinitesimal limit of the scale ϵ; a fixed value of $\epsilon = 1/2$ suffices. This is because the black and white circles of the Boolean hypercube (§ 1.2)—the only entities that impart a pattern to the hypercube—can only occupy the opposite ends of the hypercube's edges, implying that $N(\epsilon)$ does not depend on the scale ϵ as long as $\epsilon \in (0, 1)$.



(a) **(b)** **(c)** **(d)**

Figure 3.3: Calculating the dimension of a generic two-symbol schema. The schema listed under each figure is highlighted in red. The dimensions continuously progress from 1 (single edge) through 1.58 (two edges) and 2 (single face) to 2.58 (two faces). The largest possible dimension for a 3-dimensional hypercube is equal to 3, and corresponds to the schema, (#,#,#), which redescribes the entire cube (not shown).

The mean amount of unified canalization, k_u, of a Boolean automaton f is given by:

$$k_u(f) = k - \frac{\sum\limits_{f_\alpha \in F} \max\limits_{\theta : f_\alpha \in \Theta_\theta} \left(d_{f''_\theta}\right)}{|F|} \tag{3.3}$$

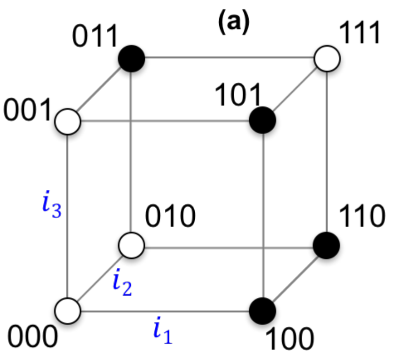
Like $k_e(f)$, lower the value of $k_u(f)$, higher the amount of canalization.

An important property of k_u is: $0 \leq k_u \leq k_e \leq k$, with $k_u < k_e$ if and only if F'' contains two-symbol schemata with $n_g > 0$ (that is, number of schemata of type (ii), $s_2 > 0$). This is due to the fact that every LUT entry of an automaton is now redescribed by a schema in F'' which is the same as, or redescribes a subset of, the schemata in F' associated with that automaton; if $F'' \equiv F'$, then $k_u = k_e$. Fig. 3.5 belows illustrates the above inequality for all possible BFs with $k = \{2, 3, 4\}$.

Moreover, the fact that the algorithm used to compute k_u misses certain schemata (explained above) implies: $k_e = k \implies k_u = k$. That is, if an automaton has no wildcard schemata with at least one wildcard symbol, then the associated F'' would be the same as F'. For example, $k_e = k_u = 2$ for the logical XOR function, since the schema $(\dot{0}, \dot{1}) \rightarrow 1$ is not considered in the calculation of k_u. Thus, the difference $k_e - k_u$ characterizes the amount of input symmetry, involving permuting wildcard symbols, in the automaton.

Fig. 3.4 illustrates a worked-out example for calculating k_u. Appendix. A describes the algorithm for calculating the k_u of a Boolean automaton, along with the example shown in Fig. 3.4, in detail.

Although a measure of input symmetry in Boolean automata (k_s) exists, as described in § 1.6.4, here we do not compare it with k_u. The reason is that the corresponding algorithms are both computationally expensive: their time complexities grow super-exponentially with k, as Fig.3.6 shows. Moreover, it makes statistical sense to compare models with a single parameter k_u, rather than two parameters k_e and k_s, to compare models with just k_e. This is because adding more parameters would always result in some model that performs at least as well as a model with



(b)

$$x = (\neg i_1 \wedge i_2 \wedge i_3) \vee (i_1 \wedge \neg i_2) \vee (i_1 \wedge \neg i_3)$$

$$F_1'' = \{(0,1,1), (1,\dot{0},\#)\}$$
$$F_0'' = \{(1,1,1), (0,\dot{0},\#)\}$$

(c)

$f_{\boldsymbol{\alpha}}$	x	Redescribing f_θ''	$\max\left(d_{f_\theta''}\right)$
(0,0,0)	0	$\{(0,\dot{0},\#)\}$	1.58
(0,0,1)	0	$\{(0,\dot{0},\dot{\#})\}$	1.58
(0,1,0)	0	$\{(0,\dot{0},\dot{\#})\}$	1.58
(1,1,1)	0	$\{(1,1,1)\}$	0
(0,1,1)	1	$\{(0,1,1)\}$	0
(1,0,0)	1	$\{(1,\dot{0},\dot{\#})\}$	1.58
(1,0,1)	1	$\{(1,\dot{0},\dot{\#})\}$	1.58
(1,1,0)	1	$\{(1,\dot{0},\dot{\#})\}$	1.58

(d)

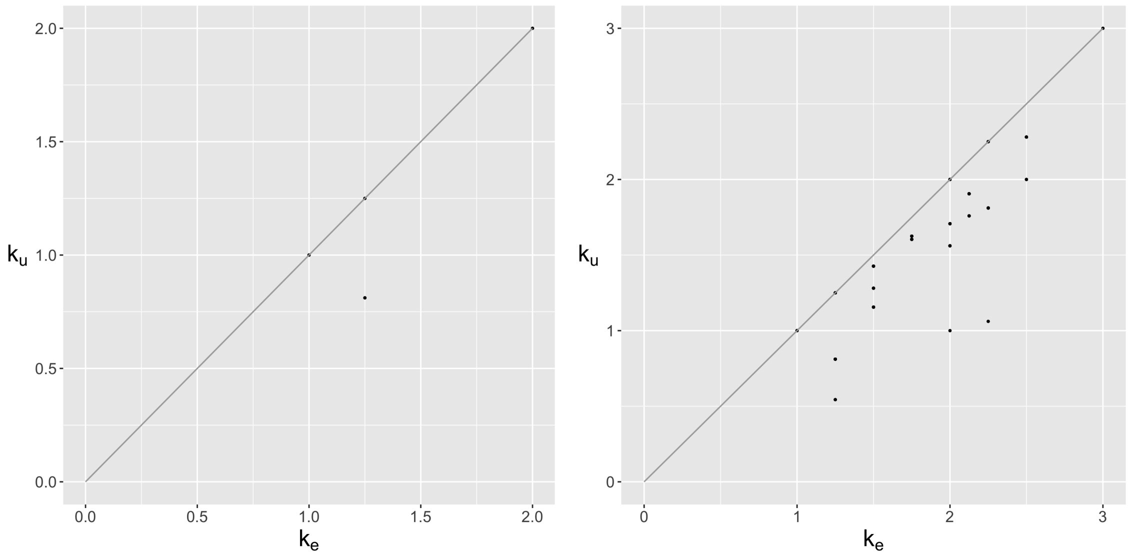
$$k_u(f) = 3 - \frac{9.51}{8} = 1.81$$

$k_e(f) = 2.25$ (for reference)

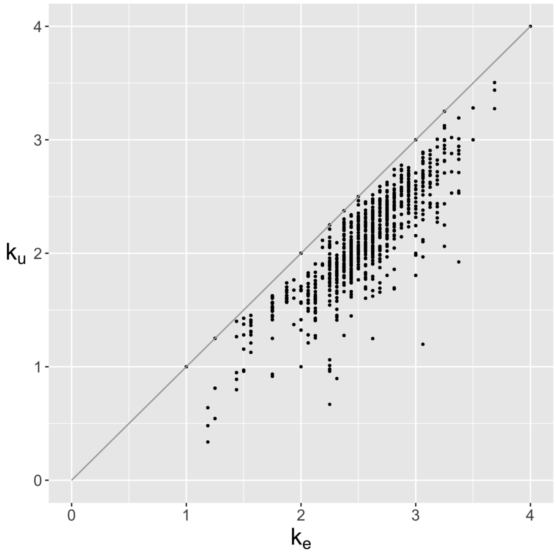
Figure 3.4: Computing the mean unified canalization of a Boolean automaton. Shown is the same BF of Fig. 1.6. **(a):** The hypercube representation of the BF (repeated for clarity). **(b):** The logic formula and the sets of redescribing two-symbol input schemata. **(c):** Sets of redescribing two-symbol schemata and the dimension of the largest redescribing schema per LUT entry. **(d)**: The numerator in the fraction term of k_u is the total of the last column in (c).

fewer parameters. Yet, as we show below, a two-parameter model consisting of k_e and a surrogate measure of k_s yields insights into the relative contributions of the two forms of canalization to the prediction of criticality.

More importantly, the fact that k_u is typically lower than k_e means that the automaton is built of even fewer "building blocks" (§ 2.1) than what k_e would suggest. Therefore, the random flip of one or more inputs is even less likely to flip the automaton's output state than what k_e would suggest. This leads to the hypothesis that the mean amount of unified canalization $\langle k_u \rangle$ of a RBN is a better predictor of dynamical regime than the mean effective connectivity $\langle k_e \rangle$, which we test next.



(a) $k = 2$ **(b)** $k = 3$



(c) $k = 4$

Figure 3.5: Illustration of the relation $k_u \leq k_e$, based on the k_e and k_u of all possible non-constant Boolean functions with $k = \{2, 3, 4\}$. The grey line is the reference $k_e = k_u$.

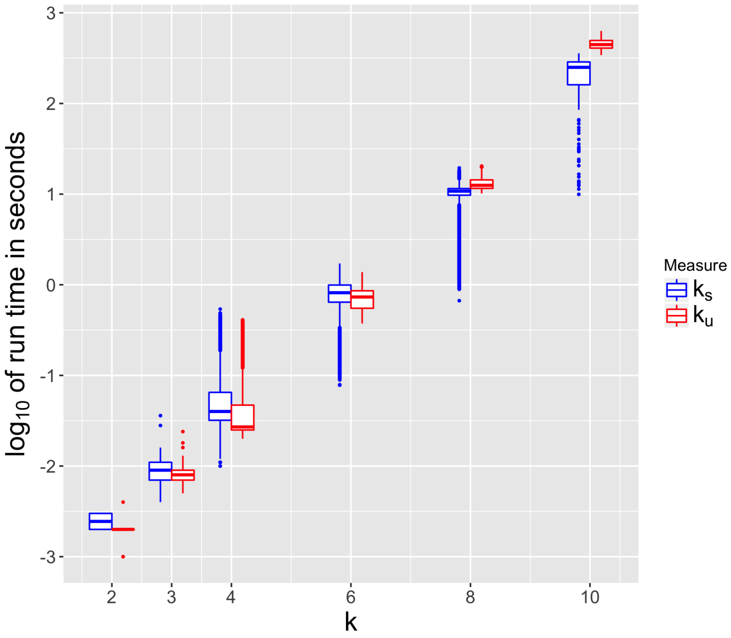


Figure 3.6: Comparison of $\log_1 0$ of run times in seconds of the algorithms that compute k_s and k_u of all possible non-constant Boolean automata with $k = \{2, 3, 4\}$ with $p \leq 0.5$, and random samples of automata with $k \in \{6, 8, 10\}$, totaling 55571 automata. Both algorithms were run on the same set of automata, written in the same language (R), and run on the same machine. As expected, the run time of the k_u algorithm grows faster than that of k_s as k grows, because the former typically detects more schemata than the latter (see main text).

3.2 Prediction of Dynamical Regime with Unified Canalization

We follow a computational approach to compare $\langle k_e \rangle$ and $\langle k_u \rangle$ as predictors of criticality of RBNs:

1. Generate RBN ensembles parameterized by both (k_e, p) and (k_u, p) over an appropriate range of values.
2. Compute their dynamic regimes based on D_c (§ 1.5.2).
3. Evaluate classification performance of statistical models with said parameters as predictors of dynamical regime.

3.2.1 RBN Ensemble Generation

The experimental setup starts by generating of RBN constrained by parameter set (k, k_e, k_u, p), so that each ensemble has both (k, k_e, p) and (k, k_u, p) parameters associated with it. The procedure for constructing an RBN with any given set of parameters is the same as described in § 2.2.1. As before, we used a 'catalog' from which the Boolean automata that make up the RBNs were picked. The catalog utilized for the experiments described below consists of: (1) the original set of automata described in § 2.2.1 with the corresponding values of k_u included; and (2) fresh sets of automata generated with $k \in \{10, 12\}$ for which both k_e and k_u were computed.

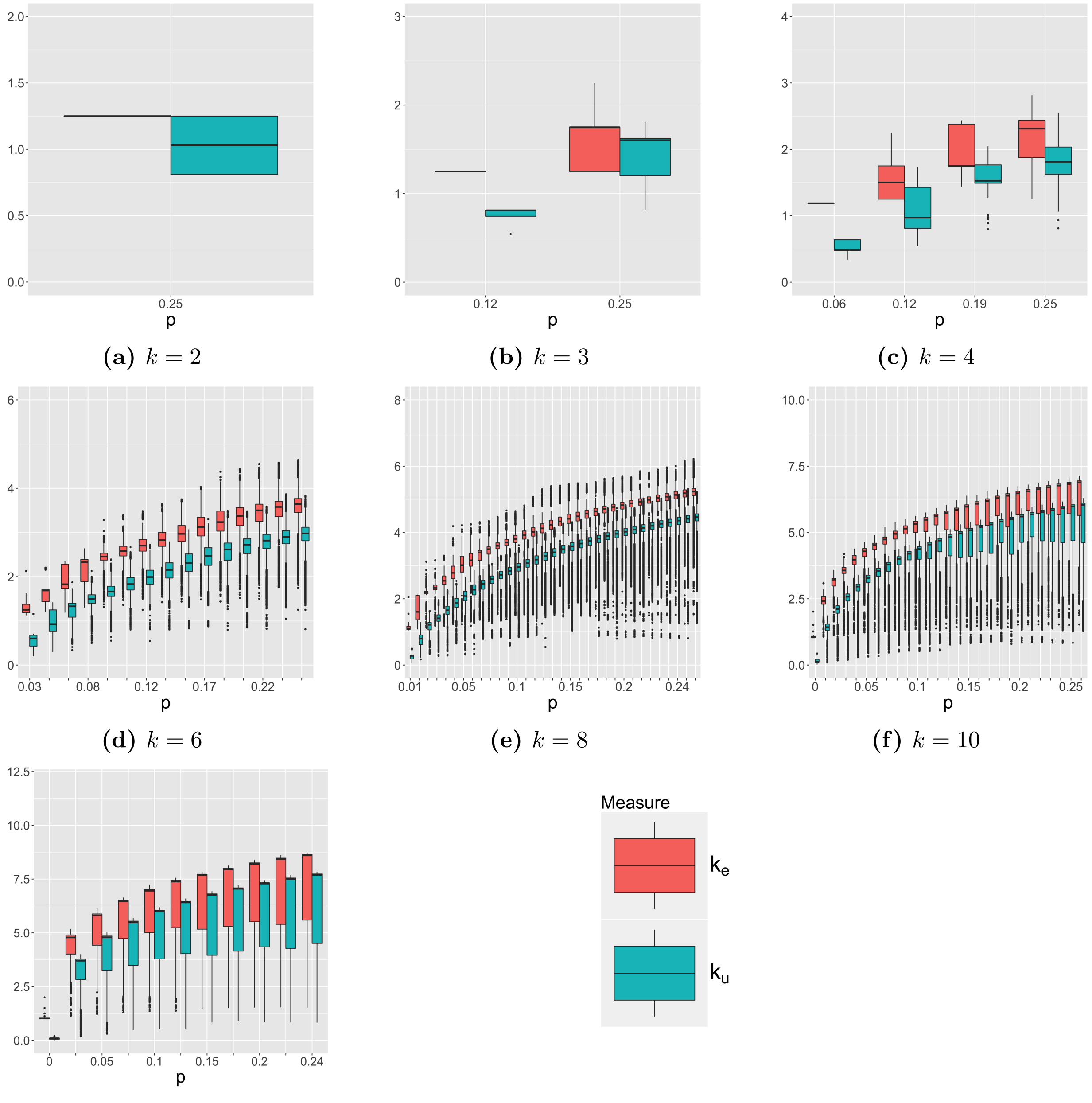
Fig. 3.7 summarizes the distributions of k_u alongside k_e of the automata in the updated catalog. As with the original catalog, the automata were sampled both randomly and with a GA (as described in § 2.2.1) for $k = 10$. For $k = 12$, however, random sampling swept so small a range of k_e and k_u that the entire set of automata for this k was generated using a GA.

The following values of the ensemble parameters were considered: $N = 100$ for all ensembles, $k \in \{2, 3, 4, 6, 8, 10, 12\}$, $p = (0, 0.26]$ with $\Delta p = 0.01$ at a minimum for all k except for $k = 12$

where $\Delta p = 0.025$, $k_e = [1, k]$ and $k_u = [1, k]$ with $\Delta k_e \approx \Delta k_u \approx 0.5$. The maximum of p was fixed at 0.26 for two reasons: (1) we observed that RBNs typically tend to be chaotic for $p > 0.26$ (Fig. 2.4(a)) even when k_e is minimal (Fig. 2.4(b)); that is, only about 0.4% of RBNs with $p > 0.26$ are stable (a proportion too small to significantly influence statistical analysis) regardless of k of k_e, hence we expected a similar behavior by k_u; (2) the computational costs of calculating k_e and k_u soars with k (Fig. 3.6).

During the construction of an RBN, the k_u of each node in an RBN was chosen from a small range of values available in the catalog, $[k_u - \Delta k_u/2, k_u + \Delta k_u/2]$, with mean k_u. By sweeping the ensemble parameters in this way, we generated a set of 100 networks for each parameter combination, resulting in the generation of a RBN set with a total of 251,467 samples, whose dynamical regimes were individually measured (§ 1.5.2).

In the analysis that follows, we will refer to the individual BNs rather than ensembles. The parameters that identify an individual BN in an ensemble are the corresponding mean values $\langle k \rangle$, $\langle k_e \rangle$, $\langle p \rangle$ and $\langle k_u \rangle$. As before, $\langle k \rangle = k$ and $\langle p \rangle = p$, that is, the k and p of all nodes in a BN are the same and equal to the associated ensemble parameter, while its $\langle k_e \rangle$ and $\langle k_u \rangle$ may be slightly different from the associated ensemble parameters k_e and k_u. In summary, we identify a BN by its k, p $\langle k_e \rangle$ and $\langle k_u \rangle$.



(a) $k = 2$ **(b)** $k = 3$ **(c)** $k = 4$

(d) $k = 6$ **(e)** $k = 8$ **(f)** $k = 10$

(g) $k = 12$

Figure 3.7: Effective connectivity (k_e) and unified canalization (k_u) of the automata in the updated catalog used in the experiments. Distributions of k_e (red) and k_u (green) per $p \in (0, 0.26]$ for $k \in \{2, 3, 4, 6, 8, 10, 12\}$. For each k, p is swept in steps of at least 0.01, except for $k = 12$ where it is swept in steps of 0.025. The legend at the bottom center

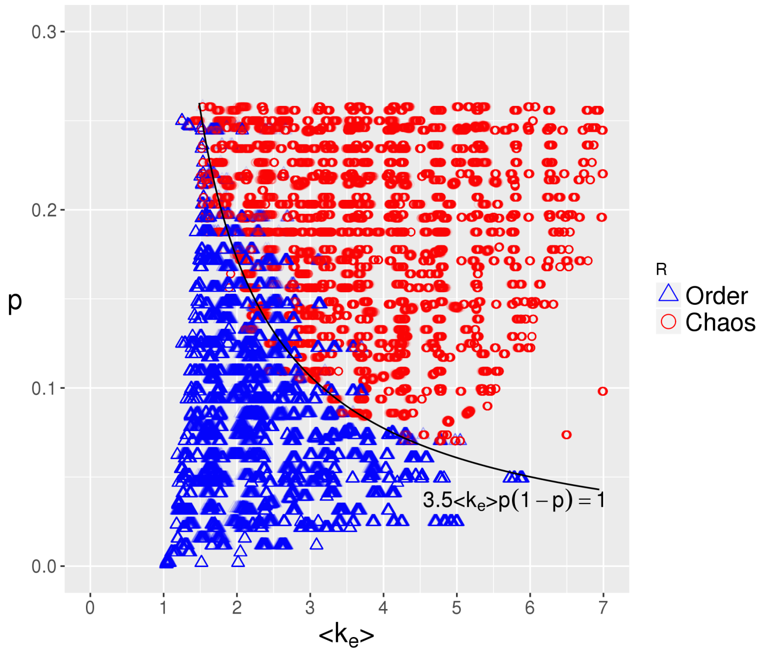
3.3 Results of Experiments

We analyze the data obtained from the ensemble experiments described above to answer the following questions:

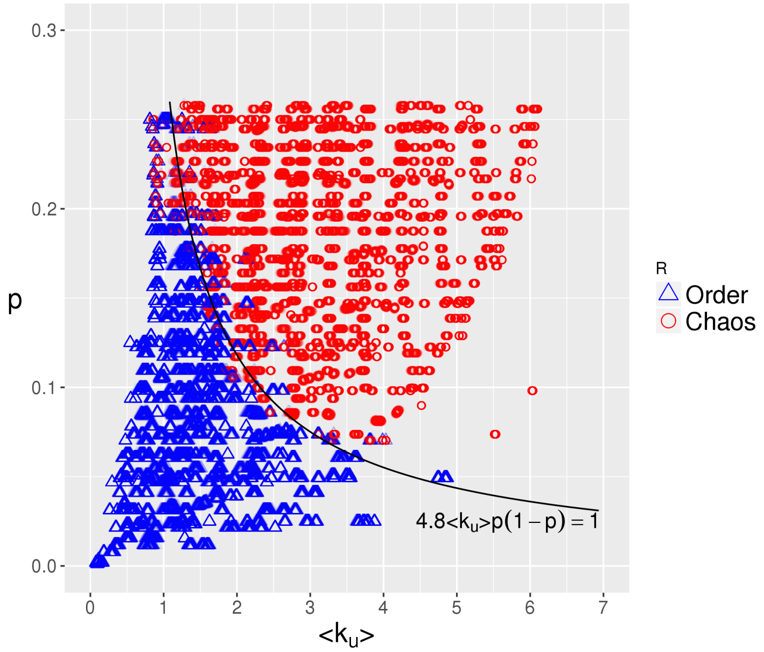
- How well do distinct statistical models based on BN parameters involving either $\langle k_e \rangle$ or $\langle k_u \rangle$ predict the observed dynamical regime?
- Are models utilizing $\langle k_u \rangle$ in place of $\langle k_e \rangle$ better predictors of dynamical regime? Does it depend on k?
- What is the best model of criticality that involves $\langle k_u \rangle$?

We formulate the above questions as a binary classification problem, and approach it in the following way. We evaluate the classification and regression performances of statistical models of varying complexities to select the best model, and then we estimate the generalization (classification and regression) performance of the best model via cross-validation.

The dynamical regime of every RBN sample is shown in Fig. 3.8 separately in the $\langle k_e \rangle$-p and $\langle k_u \rangle$-p spaces; Fig. 3.9 shows the same, except in aggregates for better visualization. As before, only the ordered and chaotic regimes are displayed in all figures, as the critical regime is combined with the former (almost no RBNs with D_c exactly equal to 1 were found). We observe that larger values of $\langle k_e \rangle$ or $\langle k_u \rangle$ tend to push the RBNs toward chaos, as expected.

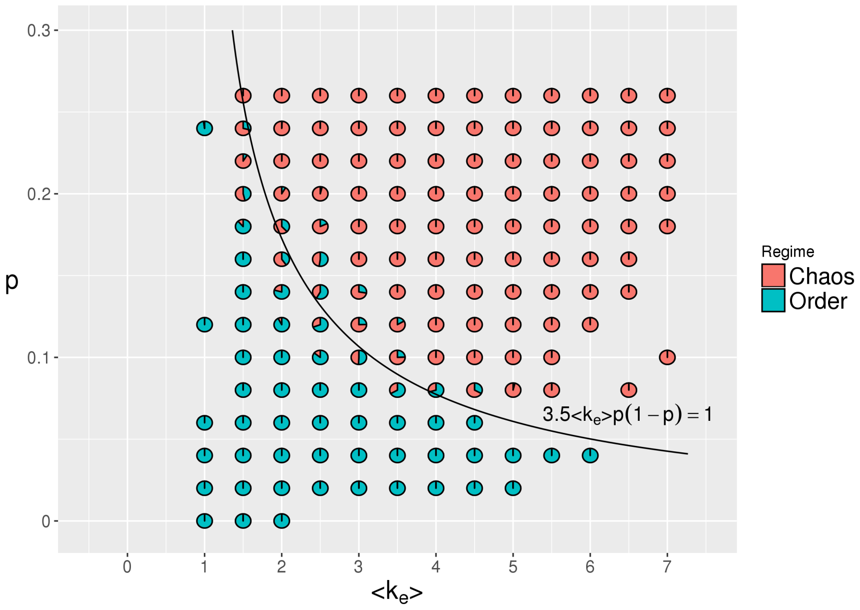


(a) $\langle k_e \rangle$-p space.

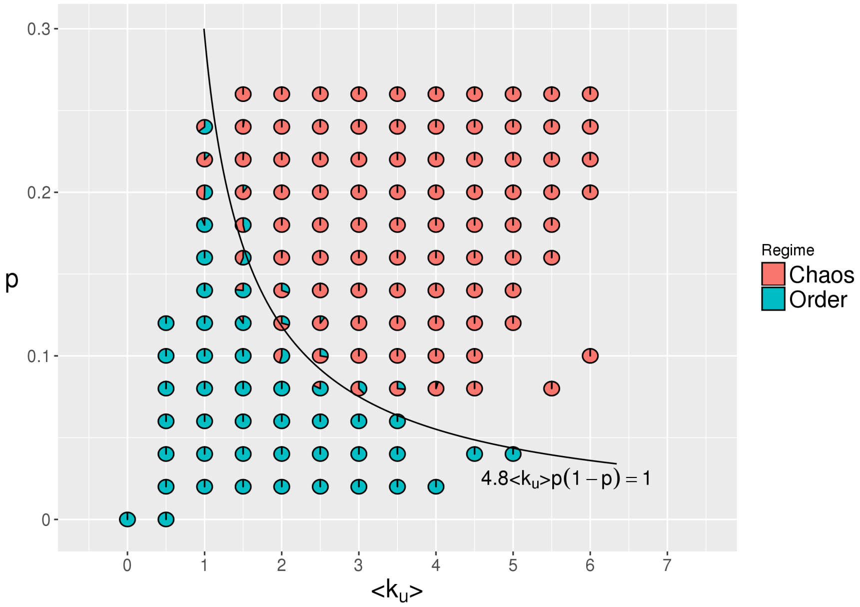


(b) $\langle k_u \rangle$-p space.

Figure 3.8: Phase diagrams in the $\langle k_e \rangle$-p and $\langle k_u \rangle$-p spaces showing the dynamical regime of every RBN sample; blue triangles represent RBNs with ordered dynamics, and red circles chaotic dynamics. Critical boundaries displayed for Model class (3), which is also the simplest model class with maximal classification performance (see main text).



(a) $\langle k_e \rangle$-p space.



(b) $\langle k_u \rangle$-p space.

Figure 3.9: Phase diagrams in the $\langle k_e \rangle$-p and $\langle k_u \rangle$-p spaces showing the dynamical regimes of aggregates of RBN samples; blue pie slices indicate the proportion of RBNs with ordered dynamics, and red indicates chaotic dynamics. Critical boundaries displayed for Model class (3), which is also the simplest model class with maximal classification performance (see main text).

To compare the ability of unified canalization with that of effective connectivity in predicting the dynamical regime of RBNs, we consider a set of models with varying degrees of complexity. For each complexity level, we compare equivalent models for $\langle k_e \rangle$ and $\langle k_u \rangle$, where one version utilizes mean effective connectivity $\langle k_e \rangle$ and the other utilizes the mean unified canalization $\langle k_u \rangle$ instead. This allows us to quantify how much the inclusion of information about symmetry in canalization, provided by two-symbol schemata on top of input redundancy provided by wildcard schemata, alters the prediction performance of a model.

As in § 2.2.2 and for the same reasons described there, we consider only multi-linear statistical models. Also as before, all models consider the term $p(1-p)$ due to logical duality in Boolean functions.

The general form of all models is:

$$R = \text{step}(\text{logistic}(\text{Model}))$$

where the output of the logistic function is the probability of observing Class label '1' (Chaos), and the output of the step function is the predicted Class labels: '1' (Chaos) if the output of the logistic function is greater than a *threshold* (T) of 0.5; and '0' (Order) otherwise. Thus, we have both a regression model and a binary classifier; the step function turns the logistic model into a classifier.

Each Model belongs to one of the following "Model classes". A total of 120 model classes are possible that include $\langle k_e \rangle$ and $\langle k_u \rangle$ in all possible interactions with k and $p(1-p)$, of which the top 10 are listed below in increasing order of 'complexity' (defined in § 2.2.2):

1. **a)** $c_1.\langle k_e \rangle$; **b)** $c_1.\langle k_u \rangle$

2. **a)** $c_1.k.\langle k_e \rangle$; **b)** $c_1.k.\langle k_u \rangle$

3. **a)** $c_1.\langle k_e \rangle.p(1-p)$; **b)** $c_1.\langle k_u \rangle.p(1-p)$

4. **a)** $c_1.k.\langle k_e \rangle.p(1-p)$; **b)** $c_1.k.\langle k_u \rangle.p(1-p)$

5. **a)** $c_1.k + c_2.\langle k_e \rangle$; **b)** $c_1.k + c_2.\langle k_u \rangle$

6. **a)** $c_1.\langle k_e \rangle + c_2.k.\langle k_e \rangle$; **b)** $c_1.\langle k_u \rangle + c_2.k.\langle k_u \rangle$

7. **a)** $c_1.\langle k_e \rangle + c_2.p(1-p)$; **b)** $c_1.\langle k_u \rangle + c_2.p(1-p)$

8. **a)** $c_1.k + c_2.k.\langle k_e \rangle$; **b)** $c_1.k + c_2.k.\langle k_u \rangle$

9. **a)** $c_1.\langle k_e \rangle + c_2.\langle k_e \rangle.p(1-p)$; **b)** $c_1.\langle k_u \rangle + c_2.\langle k_u \rangle.p(1-p)$

10. **a)** $c_1.\langle k_e \rangle + c_2.k.p(1-p)$; **b)** $c_1.\langle k_u \rangle + c_2.k.p(1-p)$

We derive a ‘criticality model’ from each regressed model in the same way as described in § 2.2.2. As in § 2.2.2, we proceed with the statistical analysis in two consecutive steps: (1) select the best characteristic model in each Model class based on the complexity and performance of all models fit to the full dataset; and (2) estimate the generalization performance of the best model obtained in the previous step via cross-validation. We use the same set of performance measures (described below) in both steps; the difference is that they are used to measure fit to the current data in the first step, and to measure prediction with respect to new data in the second step.

The following measures [21] were used to evaluate the performance of the models: R^2, MCC and AUC (defined in § 2.2.2). MCC is a particularly appropriate measure of classification performance for unbalanced classification scenarios such as is the case here: about 63% of RBNs are in the Chaos regime. We also used the model comparison significance test, namely ‘Vuong test’ and ‘Clarke test’ (§ 2.2.2) to compare the regression models with a given Model class.

As before, we consider the performance difference between the pair of models, $x.a$ and $x.b$, in a given model class x, defined as follows:

$$\text{Performance difference} = \frac{\text{Performance(Model x.b) - Performance(Model x.a)}}{\text{Performance(Model x.a)}} \tag{3.4}$$

where, $x \in \{1, \ldots, 120\}$.

The performance scores and differences of the top 10 Model classes (listed above) are displayed in Fig. 3.10 and Fig. 3.11 respectively. Results of the model comparison significance tests are displayed in Table. 3.1, and the actual models obtained containing all the coefficients, along with the values of performance measures are displayed in Table. 3.1 for the top 10 Model classes.

We observe that model version (b), utilizing $\langle k_u \rangle$, significantly outperforms version (a), utilizing $\langle k_e \rangle$, in 8 of the top 10 model classes on almost all performance measures. In the simplest model class (1), containing only the relevant canalization parameters, the MCC of version (b) outperforms version (a) by about 9% and R^2 by about 23%.

To select the best Model class from among the full suite of 120 Model classes, we use a 'pareto front' that depicts models arranged in increasing order of complexity and performance where a model is marked if and only if its performance is greater than that of all models of lower complexity (Fig. 3.12). The pareto front shows that the third simplest class of models, namely Model class (3), yields a level of performance close to the performances of the more complex model classes, while performing significantly better than the less complex Model classes—for both the $\langle k_e \rangle$ and $\langle k_u \rangle$ versions. The MCC of Model (3.a) is 0.88, which outperforms that of (1.a) by about 74%, while it is outperformed by only about 4.5% by even the most complex Model class (120.a). Likewise, the MCC of Model (3.b) is 0.89, which outperforms that of (1.b) by about 57%, while it is outperformed by only about 2.2% by even the most complex Model class (120.b). Similar trends can be observed for

R^2 and AUC in qualitative terms. In summary, with very little addition in complexity Model class (3) achieves a significant performance gain, while even a significant amount of additional complexity over Model class (3) yields only a relatively slim gain in performance. Thus, we conclude that Model class (3) is the more characteristic class of models that is optimal in terms of both simplicity and performance[7].

Within Model class (3), we observe that even though the R^2 of the Models (3.a) and (3.b) are about the same, the MCC of the $\langle k_u \rangle$ model (3.b) is better than that of the $\langle k_e \rangle$ model (3.a) (Fig. 3.11), and (3.b) also a significantly better fit as indicated by both Vuong and Clarke tests (Table. 3.1). Thus, we conclude that $\langle k_u \rangle$ is a better predictor of dynamical regime than $\langle k_e \rangle$, albeit with fairly small performance improvement.

To estimate the robustness of the coefficients and performance scores of the models, we performed repeated random stratified-sub-sampling in-validation. As before, our sub-samples are randomly chosen subsets containing 60% of the original RBN dataset, where the proportions of RBNs in the order versus chaos regimes was preserved. A set of 100 sub-samples was thus generated for each Model class and used to fit both models in each Model class and their performances were measured. Table. 3.2 summarizes the results of in-validation. We observe that the mean performance, as well as the mean values of model coefficients obtained by fitting across sub-samples, hardly deviate from what was observed in the full RBN set (Table. 3.1). Moreover, the standard deviation is very small for both the performance measures and model coefficients. This demonstrates that the main results presented in Table. 3.1 are reliable for the parameter ranges considered here. In particular, although the mean performance scores of the $\langle k_u \rangle$ version of Model class (3) are only slightly greater than those of the $\langle k_e \rangle$ version, the difference is likely robust — a conclusion bolstered by the results of model comparison significance tests (Table. 3.1).

---

[7] Model (3.a) here is the same as Model (2.b) of § 2.2.2 which was also deemed as the best model.

Class	Model	R^2	MCC	AUC	Vuong test	Clarke test
1.a	$0.39.<k_e>$	0.3	0.53	0.85	NO	NO
1.b	$0.54.<k_u>$	0.37	0.58	0.88	YES	YES
2.a	$0.05.<k_e>.k$	0.18	0.38	0.78	NO	NO
2.b	$0.07.<k_u>.k$	0.24	0.48	0.82	YES	YES
3.a	$3.53.<k_e>.p(1-p)$	0.81	0.88	0.99	NO	NO
3.b	$4.85.<k_u>.p(1-p)$	0.81	0.89	0.99	YES	YES
4.a	$0.41.<k_e>.k.p(1-p)$	0.56	0.7	0.94	NO	NO
4.b	$0.58.<k_u>.k.p(1-p)$	0.6	0.73	0.95	YES	YES
5.a	$0.69.<k_e> + -0.03.<k_e>.k$	0.36	0.58	0.88	NO	NO
5.b	$0.97.<k_u> + -0.05.<k_u>.k$	0.42	0.63	0.9	YES	YES
6.a	$-0.05.<k_e> + 3.94.<k_e>.p(1-p)$	0.81	0.89	0.99	NO	NO
6.b	$-0.08.<k_u> + 5.57.<k_u>.p(1-p)$	0.82	0.89	0.99	YES	YES
7.a	$-0.02.<k_e> + 0.44.<k_e>.k.p(1-p)$	0.56	0.7	0.94	NO	NO
7.b	$-0.03.<k_u> + 0.61.<k_u>.k.p(1-p)$	0.6	0.73	0.95	YES	YES
8.a	$-0.01.<k_e>.k + 4.08.<k_e>.p(1-p)$	0.83	0.9	0.99	NO	YES
8.b	$-0.01.<k_u>.k + 5.58.<k_u>.p(1-p)$	0.83	0.89	0.99	NO	NO
9.a	$-0.04.<k_e>.k + 0.75.<k_e>.k.p(1-p)$	0.62	0.75	0.96	NO	NO
9.b	$-0.04.<k_u>.k + 0.99.<k_u>.k.p(1-p)$	0.66	0.78	0.97	YES	YES
10.a	$4.83.<k_e>.p(1-p) + -0.15.<k_e>.p(1-p).k$	0.85	0.9	0.99	YES	YES
10.b	$6.24.<k_u>.p(1-p) + -0.16.<k_u>.p(1-p).k$	0.83	0.9	0.99	NO	NO

Table 3.1: List of criticality model expressions with coefficients and performance scores fit to the full dataset for the top 10 Model classes. Models are arranged in increasing order of complexity from top to bottom. Each grey-white banded pairs of rows are the model-pairs to be compared. The grey models contain the parameter $\langle k_e \rangle$, while its white pair contains $\langle k_u \rangle$ in its place; all other parameters assume the same role in both models. The best characteristic Model class (3) is highlighted in red. The last two columns indicate whether a model was deemed as significantly better than the other in the Model class. If neither model is better than the other (null hypothesis is not rejected), then a 'NO' would be listed for both models, as in Models (8.a) and (8.b) under 'Vuong test'.

Class	Model	Coefficients	R^2	MCC	AUC	Vuong test	Clarke test
1.a	c1.<k_e>	0.39±0.005	0.3±0.001	0.53±0.004	0.85±0	0%	0%
1.b	c1.<k_u>	0.54±0	0.37±0.003	0.58±0.004	0.88±0.001	100%	100%
2.a	c1.<k_e>.k	0.05±0	0.18±0	0.38±0	0.78±0	0%	0%
2.b	c1.<k_u>.k	0.07±0	0.24±0	0.48±0	0.82±0.002	100%	100%
3.a	c1.<k_e>.p(1-p)	3.53±0.004	0.81±0	0.88±0	0.99±0	0%	0%
3.b	c1.<k_u>.p(1-p)	4.85±0.004	0.81±0.001	0.89±0	0.99±0	100%	100%
4.a	c1.<k_e>.k.p(1-p)	0.41±0.004	0.56±0.001	0.7±0.005	0.94±0	0%	0%
4.b	c1.<k_u>.k.p(1-p)	0.58±0.001	0.6±0	0.73±0.003	0.95±0.002	100%	100%
5.a	c1.<k_e> + c2.<k_e>.k	0.69±0; -0.03±0	0.36±0.005	0.58±0.003	0.88±0	0%	0%
5.b	c1.<k_u> + c2.<k_u>.k	0.97±0.003; -0.05±0	0.42±0	0.63±0	0.9±0	100%	100%
6.a	c1.<k_e> + c2.<k_e>.p(1-p)	-0.05±0.003; 3.94±0.007	0.81±0.005	0.89±0.005	0.99±0	0%	0%
6.b	c1.<k_u> + c2.<k_u>.p(1-p)	-0.08±0; 5.58±0.012	0.82±0	0.89±0	0.99±0	100%	100%
7.a	c1.<k_e> + c2.<k_e>.k.p(1-p)	-0.02±0; 0.44±0.004	0.56±0	0.7±0.004	0.94±0	0%	0%
7.b	c1.<k_u> + c2.<k_u>.k.p(1-p)	-0.03±0.003; 0.61±0.004	0.6±0	0.73±0	0.95±0.003	100%	100%
8.a	c1.<k_e>.k + c2.<k_e>.p(1-p)	-0.01±0; 4.08±0.005	0.83±0	0.9±0	0.99±0	4%	100%
8.b	c1.<k_u>.k + c2.<k_u>.p(1-p)	-0.01±0; 5.58±0.009	0.83±0	0.89±0.004	0.99±0	0%	0%
9.a	c1.<k_e>.k + c2.<k_e>.k.p(1-p)	-0.04±0.002; 0.75±0.002	0.62±0.005	0.75±0	0.96±0	0%	0%
9.b	c1.<k_u>.k + c2.<k_u>.k.p(1-p)	-0.04±0; 0.99±0.005	0.66±0.005	0.78±0	0.97±0	100%	100%
10.a	c1.<k_e>.p(1-p) + c2.<k_e>.p(1-p).k	4.83±0.009; -0.15±0	0.85±0	0.9±0	0.99±0	100%	100%
10.b	c1.<k_u>.p(1-p) + c2.<k_u>.p(1-p).k	6.24±0.015; -0.16±0.003	0.83±0	0.9±0	0.99±0	0%	0%

Table 3.2: List of criticality model expressions with the central tendencies and standard deviations of coefficients and performance scores obtained from repeated random stratified-sub-sampling in-validation for the top 10 Model classes. Models are arranged in increasing order of complexity from top to bottom. Each grey-white banded pairs of rows are the model-pairs to be compared. The grey models contain the parameter $\langle k_e \rangle$, while its white pair contains $\langle k_u \rangle$ in its place; all other parameters assume the same role in both models. The best characteristic Model class (3) is highlighted in red. The last two columns indicate the proportion of comparison tests where a model was deemed as significantly better than the other in the Model class.

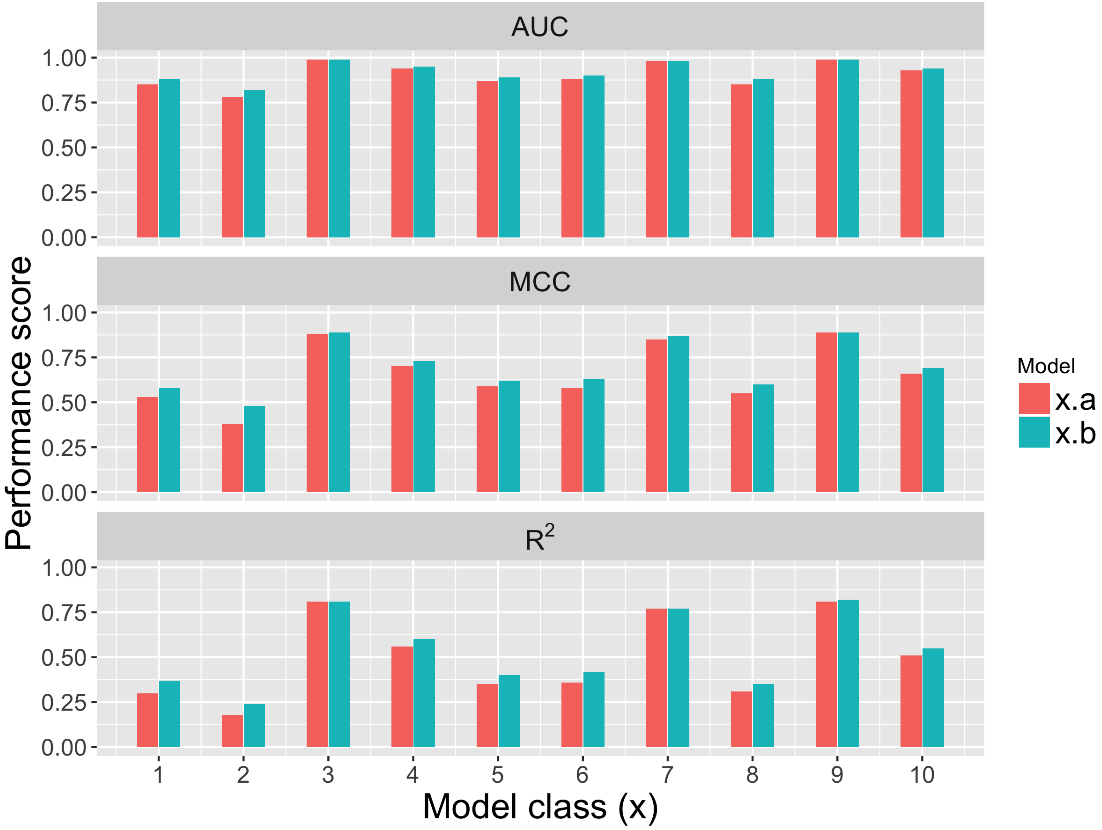


Figure 3.10: Performance of the top 10 model classes fit to the full dataset. Models are arranged in increasing order of complexity from left to right.

To make sure the results above do not derive from overfitting the data, and to estimate generalization performance on out-of-sample RBNs, we performed cross-validation on the best Model class (3). The procedure is the same as described in § 2.2.2, except it now uses Model class (3) in the training steps. The results of cross-validation are summarized in Fig. 3.13 and Table. 3.3. We observe that the cross-validated MCC and R^2 of Model (3.b) is significantly better than that of (3.a), and (3.b) is also a significantly better generalizable model than (3.a) per the model comparison significance tests. Thus, we conclude that not only Model (3.b) is a better characteristic model than (3.a), it also generalizes significantly better than (3.a). Moreover, the cross-validated performance scores of Model (3.b) are essentially the same as the ordinary performance scores, and can therefore be used as a generic model to make predictions of dynamical regimes of homogeneous

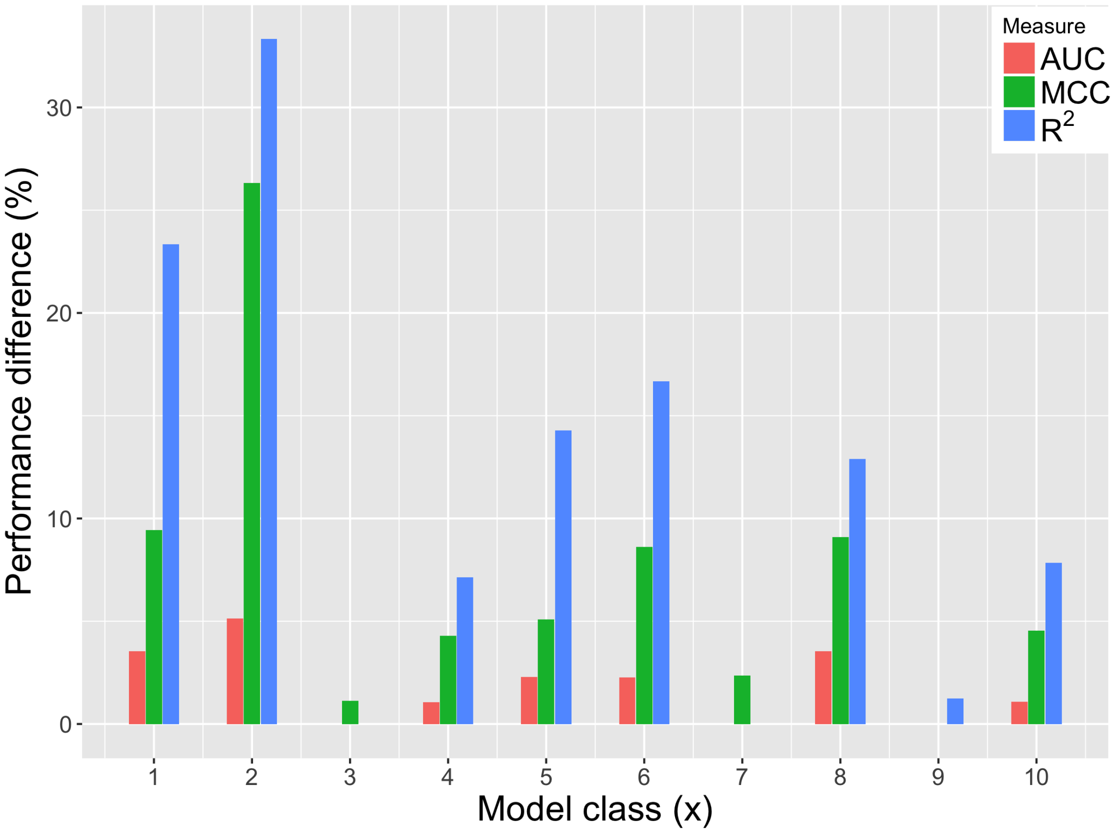


Figure 3.11: Performance differences of the top 10 model classes. Models are arranged in increasing order of complexity from left to right.

RBNs in general for the RBN parameter ranges considered here.

Since $\langle k_u \rangle$ is only slightly better in performance than $\langle k_e \rangle$, we further assessed their performance differences by fitting Model class (3) to subsets of data conditioned on k, with the goal of extrapolating whether the differences are likely to improve with more data containing larger values of k. That is, for each k, we fit *new* models, denoted as Models (3.a(k)) and (3.b(k)), and compared them using the same performance scores and difference calculation as above. The raw performance scores are displayed in Fig. 3.14 and detailed in Table. 3.4. The performance difference is calculated as follows and displayed in Fig. 3.15:

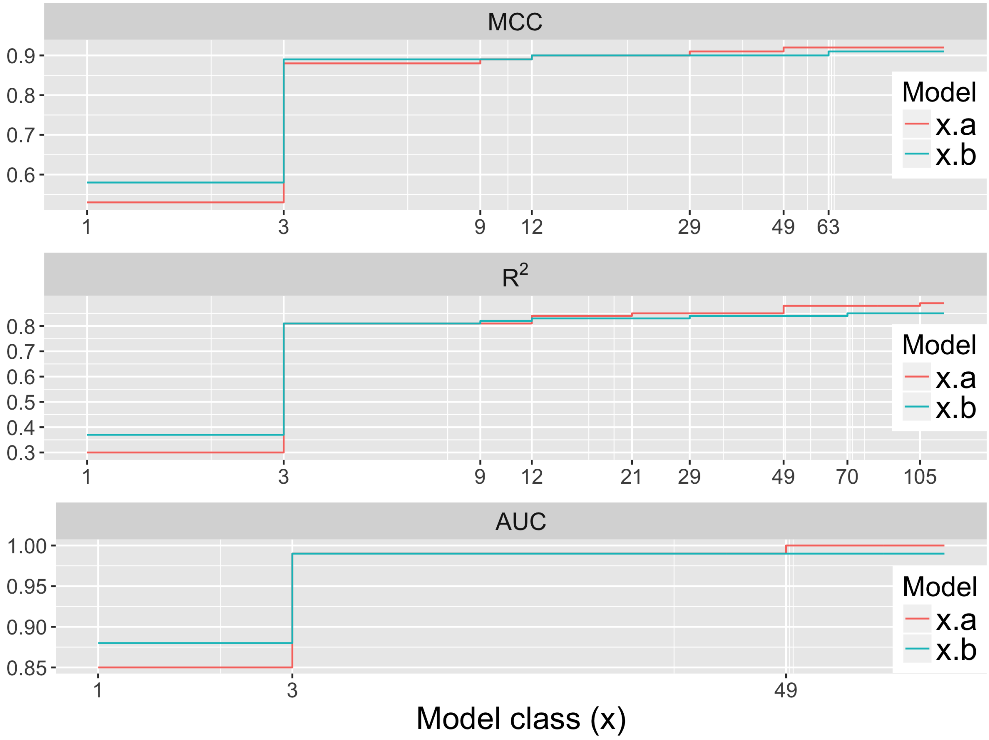


Figure 3.12: A 'pareto front' of the model complexity versus performance for the full suite of 120 model classes fit to the full dataset. Models are arranged in increasing order of complexity from left to right. A model class is marked (labeled on the axis) if and only if its performance is greater than the performances of all models of lower complexity. The horizontal axis is log-scaled just to show in better detail the performances of the simpler models.

$$\text{Performance difference} = \frac{\text{Performance(Model 3.b(k)) - Performance(Model 3.a(k))}}{\text{Performance(Model 3.a(k))}}$$

where, $k \in \{2, 3, 4, 6, 8, 10, 12\}$

We observe that, with increasing values of k, there is a systematic reduction in the performance difference in both the MCC and R^2 scores up to $k = 8$ after which with the $\langle k_u \rangle$ version begins to outperform the $\langle k_e \rangle$ version: by about 2.2% at $k = 10$ and by about 4.6% at $k = 12$ in terms of MCC. This qualitative trend is observed for R^2 as well. The results of model comparison significance tests (Table. 3.4) concur with these trends. These observations together suggest that $\langle k_u \rangle$ is likely

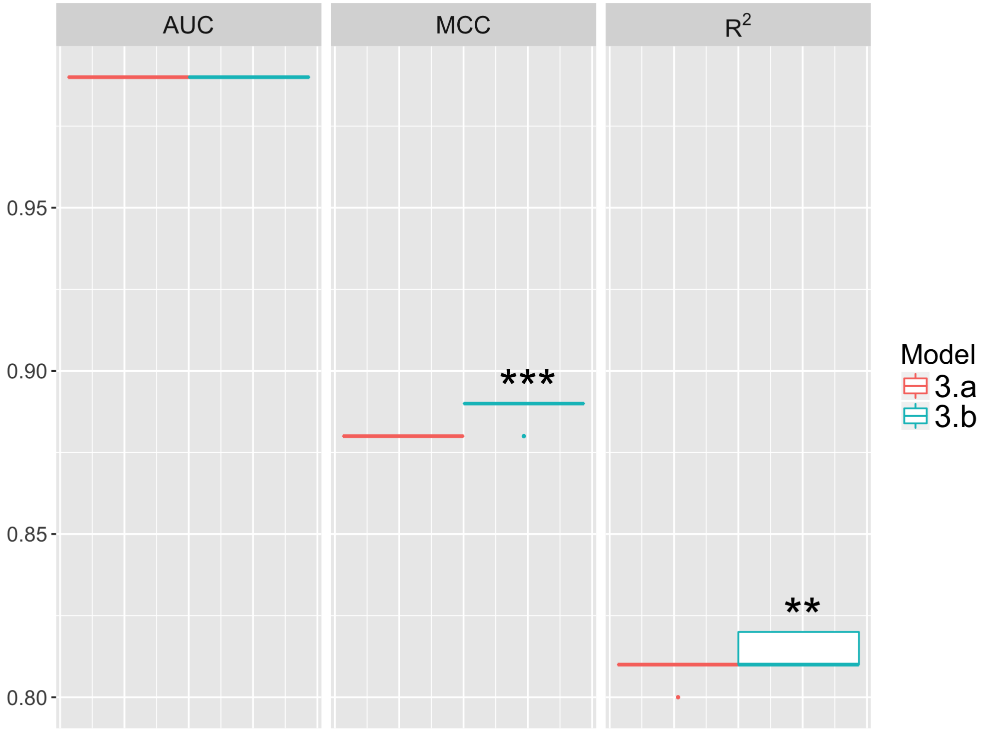


Figure 3.13: Generalization performance of Model class (3) from nested 4-fold cross-validation. The significance codes above the boxes indicate p-values ($<$ 0.001 = '***'; $<$ 0.01 = '**') obtained from a one-sided paired-sample t-test where variances of the samples were not assumed to be the same. Here, paired-sample was assumed because the two models were evaluated on the same set of 16 test folds (§ 2.2.2); and one-sided test means that the alternative hypothesis is that the mean score of Model (3.b) is greater than that of (3.a).

to become even better a predictor, continuing the trend that originated at $k = 10$, of dynamical regime than $\langle k_e \rangle$ at larger values of k.

To ensure that the above observations are not exclusive to Model class (3), even though we have deemed it to be the best[8], we computed the mean performance differences for each measure over the full suite of 120 model classes, defined as follows:

---

[8] It is possible that with more data containing RBNs with larger values of k, Model class (3) is not the best anymore.

Class	Criticality model class	R^2	MCC	AUC	Vuong test	Clarke test
3.a	c1.<k_e>.p(1-p)	0.81±0.002	0.88±0	0.99±0	0%	0%
3.b	c1.<k_u>.p(1-p)	0.81±0.005	0.89±0.002	0.99±0	93.75%	100%

Table 3.3: Central tendencies and standard deviations of the generalization performance of Model class (3) from nested 4-fold cross-validation. According to the model comparison significance tests, with respect to most if not all of the 16 test folds, Model (3.b) was significantly better than (3.a).

k	Class	Criticality model	R^2	MCC	AUC	Vuong test	Clarke test
2	3.a(2)	0.<k_e>.p(1-p)	0	0	0.5	YES	YES
2	3.b(2)	-1.18.<k_u>.p(1-p)	0	0	0.54	NO	NO
3	3.a(3)	4.05.<k_e>.p(1-p)	0.94	0.98	0.99	YES	YES
3	3.b(3)	4.51.<k_u>.p(1-p)	0.63	0.62	0.96	NO	NO
4	3.a(4)	3.92.<k_e>.p(1-p)	0.98	1	1	YES	YES
4	3.b(4)	5.33.<k_u>.p(1-p)	0.74	0.74	0.98	NO	NO
6	3.a(6)	4.04.<k_e>.p(1-p)	0.94	0.97	1	YES	YES
6	3.b(6)	5.43.<k_u>.p(1-p)	0.83	0.88	0.99	NO	NO
8	3.a(8)	3.85.<k_e>.p(1-p)	0.92	0.96	1	YES	YES
8	3.b(8)	5.27.<k_u>.p(1-p)	0.88	0.94	1	NO	NO
10	3.a(10)	3.36.<k_e>.p(1-p)	0.87	0.91	1	YES	NO
10	3.b(10)	4.71.<k_u>.p(1-p)	0.87	0.93	0.99	NO	YES
12	3.a(12)	2.9.<k_e>.p(1-p)	0.79	0.86	0.99	NO	NO
12	3.b(12)	4.1.<k_u>.p(1-p)	0.81	0.9	0.99	YES	YES

Table 3.4: List of criticality model expressions with coefficients and performance scores fit to subsets of data conditioned on k for Model class (3). The last two columns indicate the proportion of comparison tests where a model was deemed as significantly better than the other in a Model class for a given k. Highlighted in red are the k when Model (3.b(k)) begins to perform better than (3.a(k)).

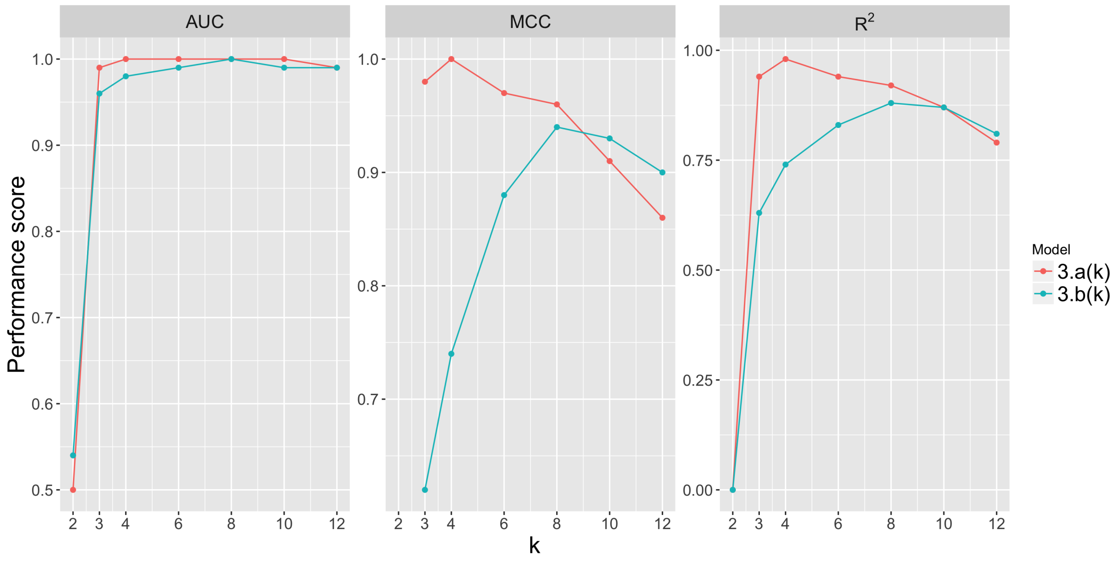


Figure 3.14: Performance scores of Models (3.a(k)) and (3.b(k)) fit to subsets of data conditioned on each $k \in \{2, 3, 4, 6, 8, 10, 12\}$.

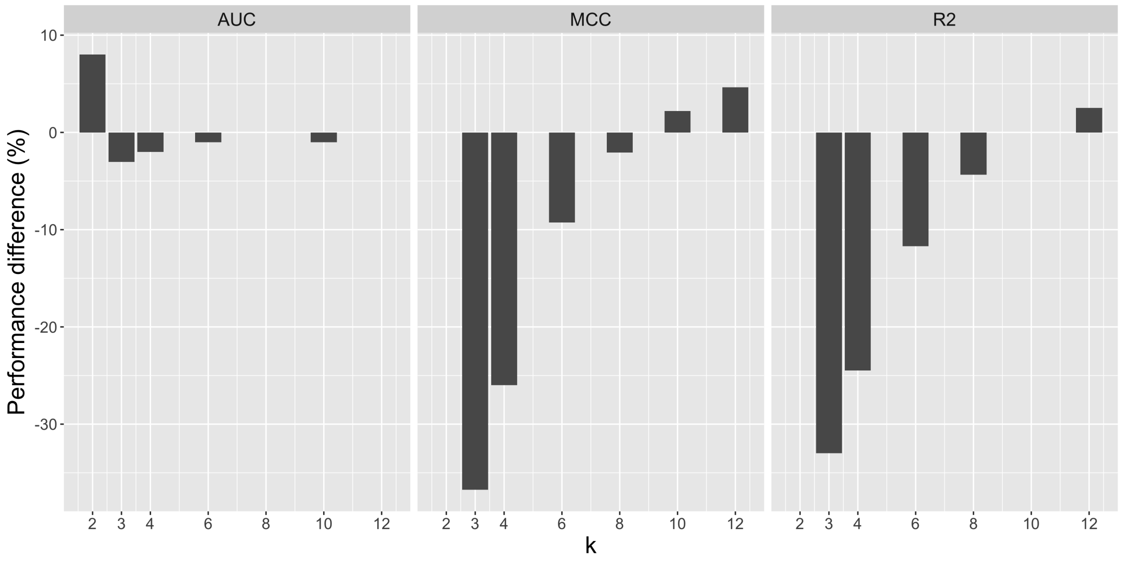


Figure 3.15: Performance difference between Models (3.a(k)) and (3.b(k)) fit to subsets of data conditioned on each $k \in \{2, 3, 4, 6, 8, 10, 12\}$.

$$\text{Mean performance difference} = \frac{1}{120} \cdot \sum_{x=1}^{120} \frac{\text{Performance(Model x.b(k)) - Performance(Model x.a(k))}}{\text{Performance(Model x.a(k))}}$$

where, $k \in \{2, 3, 4, 6, 8, 10, 12\}$

The mean performance differences are displayed in Fig. 3.16. The proportions of significantly better models in a given Model class deemed than the other in the same class, as deemed by the Vuong and Clarke tests, for each k for the full suite of 120 Model classes are displayed in Fig. 3.17. The qualitative trends that can be discerned from these figures are coherent with those discussed above. Together, these observations suggest that $\langle k_u \rangle$ is likely to become a better predictor than $\langle k_e \rangle$ at larger values of k irrespective of how they interact with the other RBN parameters. One possible explanation for this is that with larger k, more room is created for symmetry to play a role. This idea is supported by Fig. 3.2 which shows that schema signatures with $n_g > 0$ dominate the space of signatures at large values of k—a greater proportion of information about canalization stored in these signatures suggests more predictive information about criticality as well.

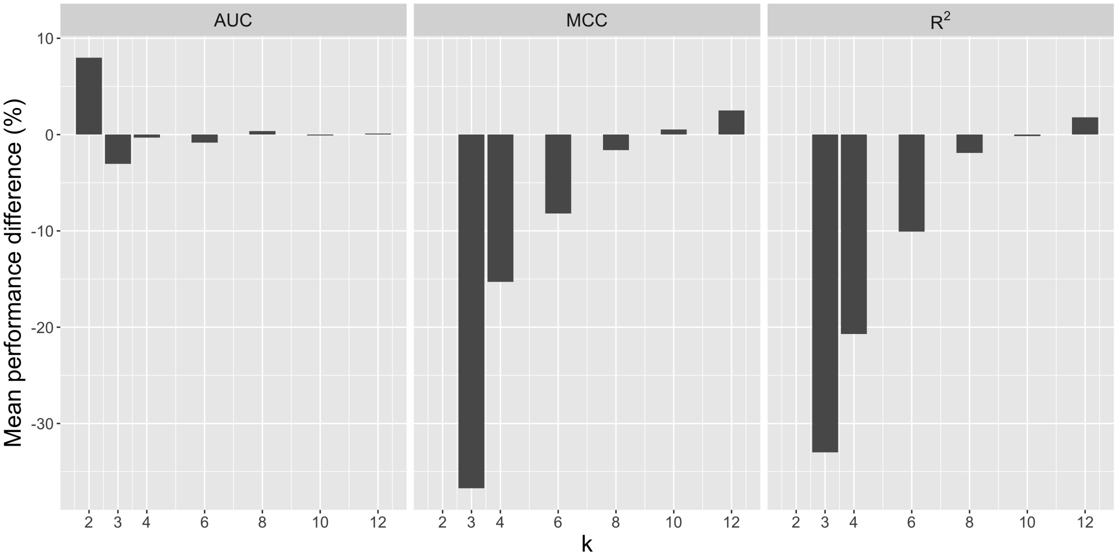


Figure 3.16: Mean performance differences computed over the full suite of 120 Model classes fit to subsets of data conditioned on k.

However, as Figs. 3.14 - 3.17 and Table. 3.4 show, $\langle k_e \rangle$ significantly outperforms $\langle k_u \rangle$ when $k \leq 8$, even though the latter contains more information about canalization. This suggests that k_u may not be an appropriate measure of canalization for RBNs with relatively smaller values of k. To test this possibility, we dissociate the two forms of canalization, namely effective connectivity and

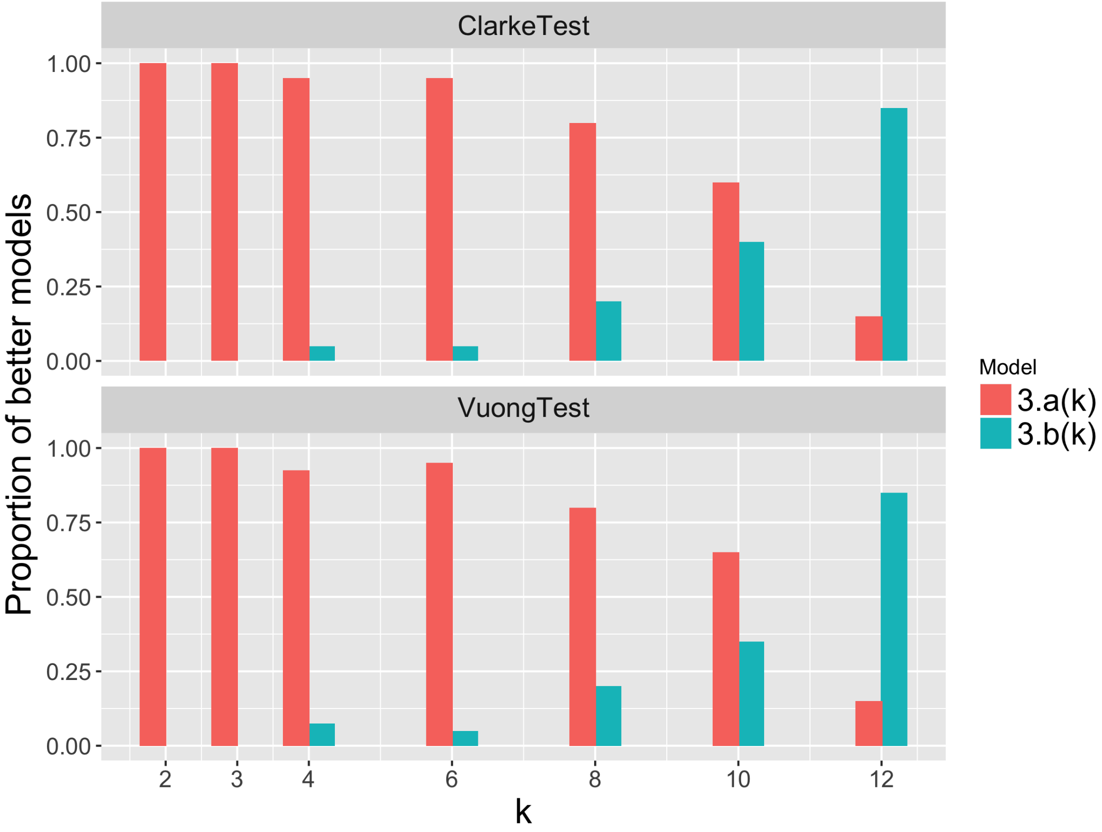


Figure 3.17: Mean proportions of models significantly better than the other in a Model class, as computed over the full suite of 120 Model classes fit to subsets of data conditioned on k.

input symmetry (with permuting wildcards), from k_u (as explained in § 3.1) and consider a model with them as independent predictors. That is, we define a measure of input symmetry[9], as follows:

$$k_y = k_u - k_e$$

and, consider the following Model class:

$$\textbf{3.c)}\quad c_1.\langle k_e\rangle.p(1-p) + c_2.\langle k_y\rangle.p(1-p)$$

---

[9] Not the same as k_s but correlates with it.

Fig. 3.18 depicts the R^2 and MCC scores of Models (3.a) and (3.b) fit to the full dataset conditioned on various values of k, and Table. 3.5 depicts the results of ANOVA comparison of the two regression models, based on a likelihood ratio test (LRT)[10] follows a Chi-squared distribution under the null hypothesis that the two models are statistically equivalent [102]. We observe that, Model (3.c), unlike (3.b), performs as well as (3.a) up until $k = 8$, with the exception of a slight drop of about 1% in MCC at $k = 6$. Beyond $k = 8$, Model (3.c) begins to outperform (3.a) with a gain of about 2.5% in MCC and about 4.7% in R^2 at $k = 12$. This result should not be unexpected since (3.a) is a nested model of (3.c). Furthermore, Model (3.c) is significantly better than (3.a) starting $k = 4$; and, as k increases, the significance also increases as reflected by the corresponding Chi-squared values[11].

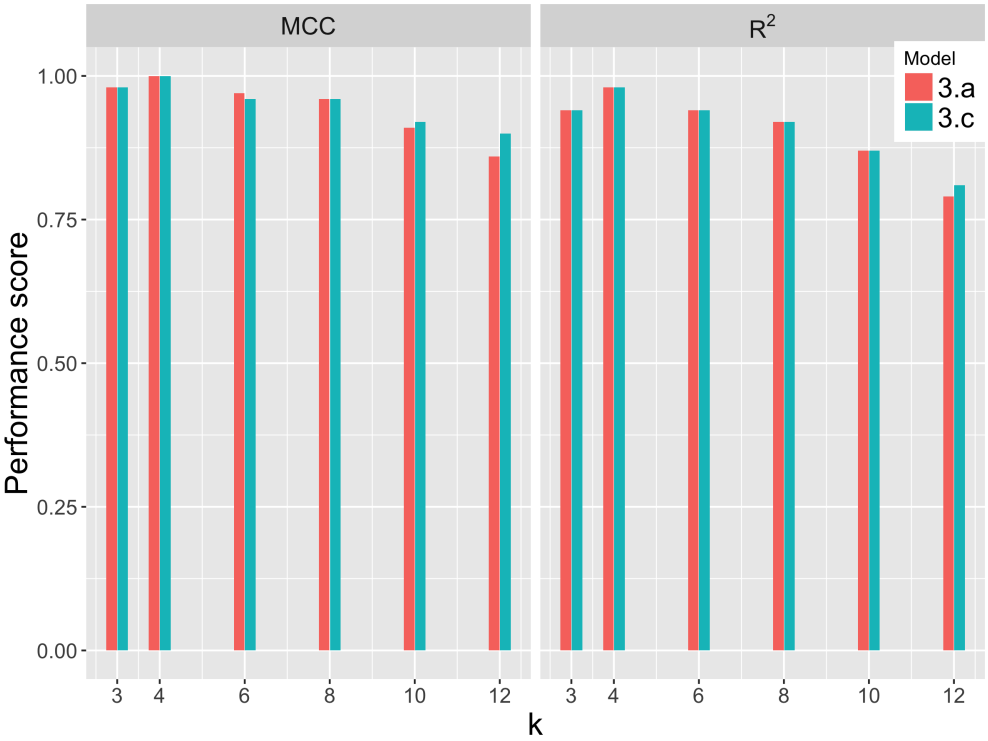


Figure 3.18: Performance of Model class (3.c) fit to subsets of data conditioned on k.

---

[10] -2 times the sum of the logarithm of the likelihood ratios of (3.a) to (3.c) for each data sample

[11] One reason why the increase in significance is not reflected in the difference in R^2 values is because the latter is based on a comparison of the given Model with an intercept-only null Model, whereas LRT is based on a direct comparison of the two given models which picks up more subtle differences between them.

k	Criticality models		Likelihood ratio test		R^2		MCC	
	3.a	3.c	Chisq	p-value [Pr(>Chisq)]	3.a	3.c	3.a	3.c
3	$4.05.<k_e>.p(1-p)$	$4.05.<k_e>.p(1-p) + 0.01.<k_y>.p(1-p)$	0	0.934823673	0.94	0.94	0.98	0.98
4	$3.92.<k_e>.p(1-p)$	$4.03.<k_e>.p(1-p) - 0.39.<k_y>.p(1-p)$	13.93	0.000189964	0.98	0.98	1	1
6	$4.04.<k_e>.p(1-p)$	$4.13.<k_e>.p(1-p) - 0.34.<k_y>.p(1-p)$	45.88	1.26E-11	0.94	0.94	0.97	0.96
8	$3.85.<k_e>.p(1-p)$	$4.<k_e>.p(1-p) - 0.52.<k_y>.p(1-p)$	90.02	2.36E-21	0.92	0.92	0.96	0.96
10	$3.36.<k_e>.p(1-p)$	$3.9.<k_e>.p(1-p) - 1.9.<k_y>.p(1-p)$	490.05	1.39E-108	0.87	0.87	0.91	0.92
12	$2.9.<k_e>.p(1-p)$	$4.12.<k_e>.p(1-p) - 4.16.<k_y>.p(1-p)$	1009.67	1.42E-221	0.79	0.81	0.86	0.9

Table 3.5: Performance of Model class (3.c) fit to subsets of data conditioned on k. Significant p-values (< 0.001) are highlighted in red.

From the above observations, we conclude that the unified k_u is not appropriate measure of canalization for small values of k insofar as the prediction of dynamical regime is concerned. This is because dissociating the two forms of canalization results in a better criticality model where they contribute independently in different proportions to the prediction, whereas in the unified-parameter model they are forced to contribute equally resulting in an overall prediction loss (Table. 3.4). However, at a relatively high value of $k = 12$, the two-parameter Model (3.c) performs equally well as the unified-parameter Model (2.b) both in terms of R^2 and MCC (Table. 3.5 and 3.4). Whether k_u remains an appropriate measure of canalization for $k > 12$ in the above sense is an open question that could be answered in future work.

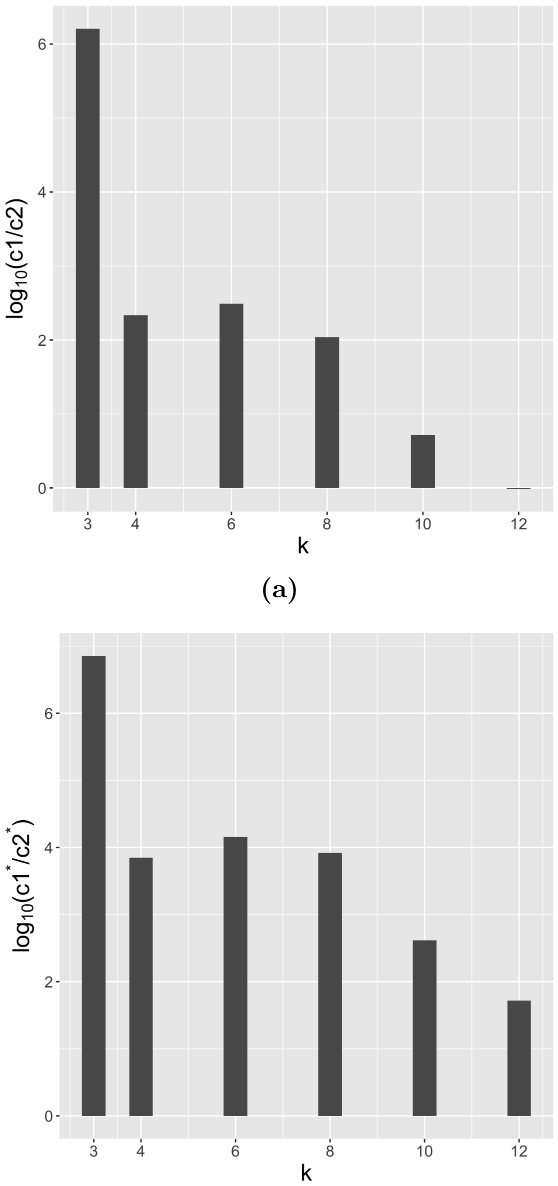
Next, we assess the relative importance of the $\langle k_e \rangle$ and $\langle k_y \rangle$ terms in Model (3.c). Fig. 3.19(a) depicts the logarithm of the ratio c_1/c_2 of the corresponding coefficients, and Fig. 3.19(b) depicts the same but with respect to *standardized* coefficients c_1^* and c_2^* calculated based on a standardization method described by Menard [92]. The purpose of standardization is to make the predictors in a regression model more comparable. In this case, one standard deviation in $\langle k_e \rangle \langle p \rangle (1 - \langle p \rangle)$ translates into c_1^* standard deviations in the dependent variable[12], and one standard deviation in $\langle k_y \rangle \langle p \rangle (1 - \langle p \rangle)$ translates into c_2^* standard deviations in the dependent variable. In both cases, we observe that the relative importance of the $\langle k_e \rangle$ term decreases as k increases. That is, $\langle k_y \rangle$ term becomes increasingly more important as k increases, and at $k = 12$ it becomes as important as $\langle k_e \rangle$ per c_1/c_2, and almost as important per c_1^*/c_2^*.

The individual importances of the $\langle k_e \rangle$ and $\langle k_y \rangle$ terms can be assessed using: (1) a 'Wald' test [102] where the square of the ratio of corresponding coefficient and its standard error follows a Chi-squared distribution under the null hypothesis that the corresponding is equal to 0; and (2) an LRT test (as above) comparing the full Model with a partial Model where a term is dropped. Table. 3.6 shows the results of these tests. We observe in both tests that the test statistic (Chi-squared values) for the $\langle k_e \rangle$ term increases as k rises up to $k = 10$ after which it drops by about 50%; whereas, for the $\langle k_y \rangle$ term the test statistic steadily increases as k increases, doubling at $k = 12$ as at $k = 10$. The relative importance of the terms is indicated by the ratio of the Chi-squared values of the $\langle k_e \rangle$ term to that of the $\langle k_y \rangle$ term which shrinks as k increases (Fig. 3.20). Concurring with our earlier observations, these results show that the $\langle k_y \rangle$ term becomes increasingly more important as k increases.

Together the above observations further clarify our earlier result that $\langle k_u \rangle$ becomes a better predictor as k increases: it is the input symmetry component of $\langle k_u \rangle$ that contributes more to this phenomenon than effective connectivity.

---

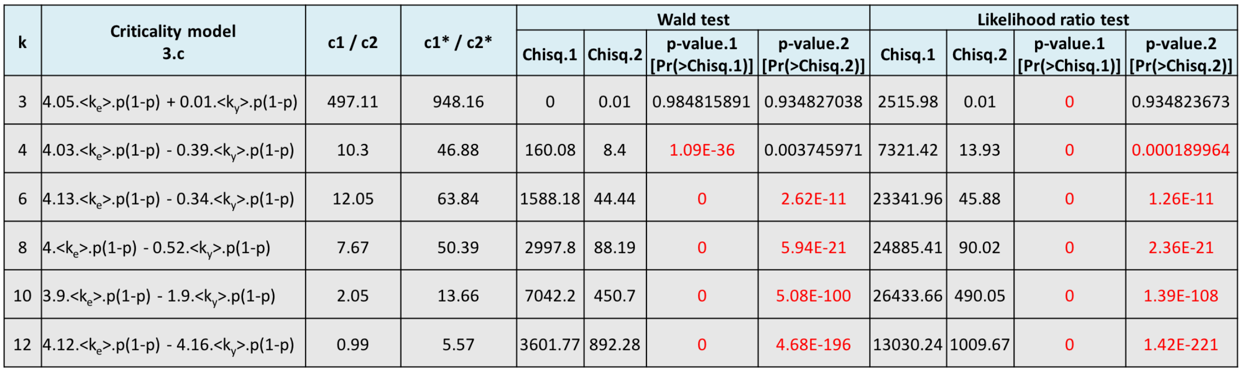
[12] In particular, a logit transformation of the dependent variable for logistic regression models.



(a)

(b)

Figure 3.19: The relative importance of the terms in Model class (3.c).



k	Criticality model 3.c	c1 / c2	c1* / c2*	Wald test				Likelihood ratio test			
				Chisq.1	Chisq.2	p-value.1 [Pr(>Chisq.1)]	p-value.2 [Pr(>Chisq.2)]	Chisq.1	Chisq.2	p-value.1 [Pr(>Chisq.1)]	p-value.2 [Pr(>Chisq.2)]
3	4.05.<k_e>.p(1-p) + 0.01.<k_y>.p(1-p)	497.11	948.16	0	0.01	0.984815891	0.934827038	2515.98	0.01	0	0.934823673
4	4.03.<k_e>.p(1-p) - 0.39.<k_y>.p(1-p)	10.3	46.88	160.08	8.4	1.09E-36	0.003745971	7321.42	13.93	0	0.000189964
6	4.13.<k_e>.p(1-p) - 0.34.<k_y>.p(1-p)	12.05	63.84	1588.18	44.44	0	2.62E-11	23341.96	45.88	0	1.26E-11
8	4.<k_e>.p(1-p) - 0.52.<k_y>.p(1-p)	7.67	50.39	2997.8	88.19	0	5.94E-21	24885.41	90.02	0	2.36E-21
10	3.9.<k_e>.p(1-p) - 1.9.<k_y>.p(1-p)	2.05	13.66	7042.2	450.7	0	5.08E-100	26433.66	490.05	0	1.39E-108
12	4.12.<k_e>.p(1-p) - 4.16.<k_y>.p(1-p)	0.99	5.57	3601.77	892.28	0	4.68E-196	13030.24	1009.67	0	1.42E-221

Table 3.6: Individual importance of the terms in Model class (3.c) fit to subsets of data conditioned on k. Significant p-values (< 0.001) are highlighted in red.

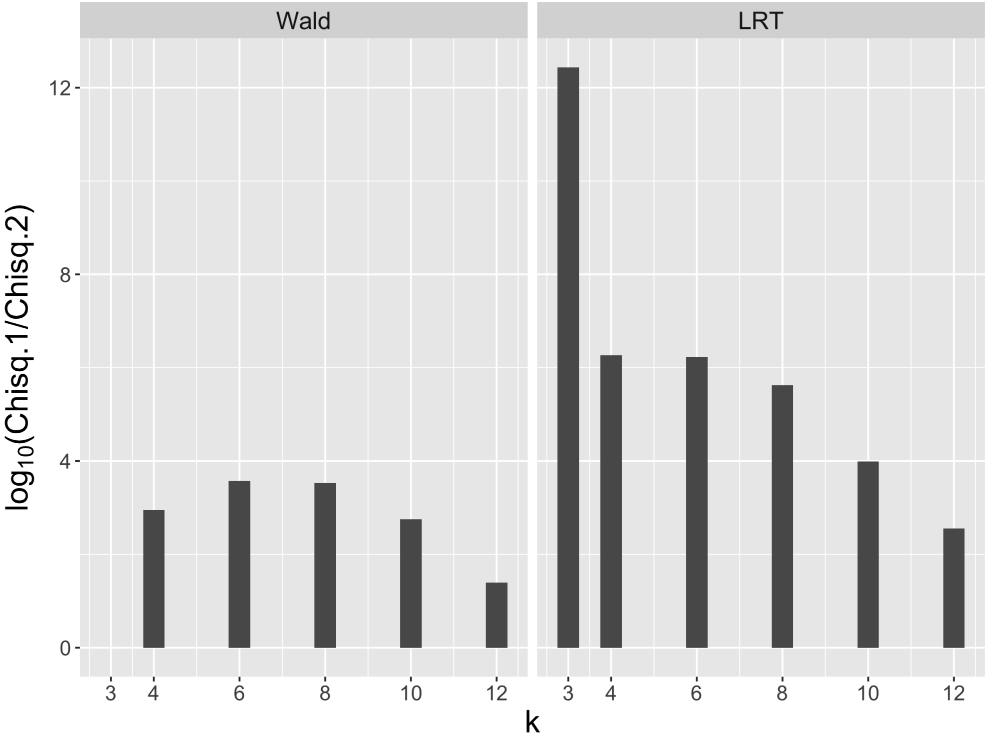
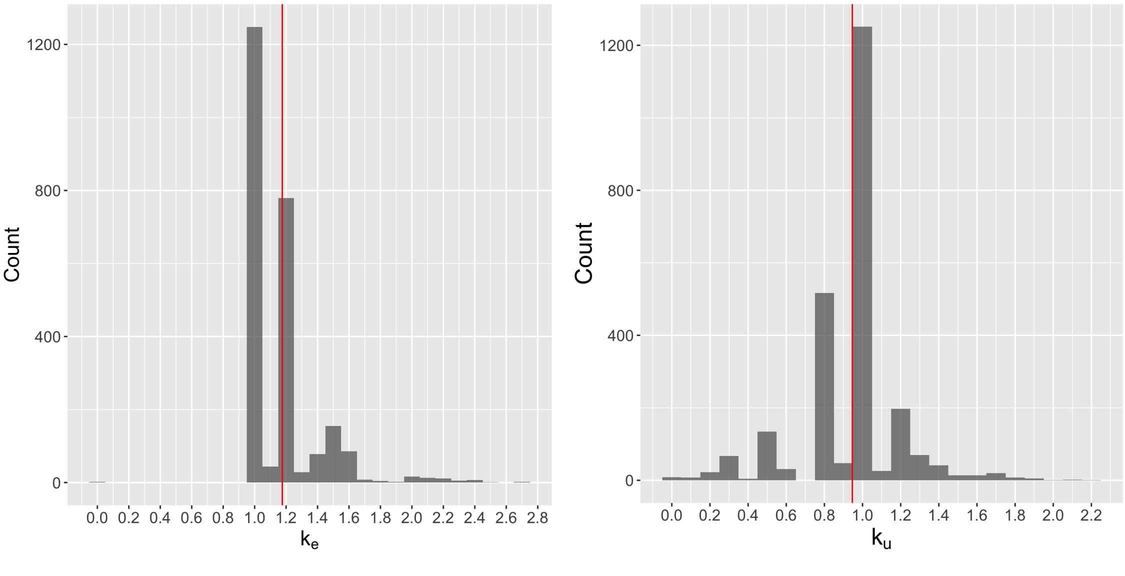


Figure 3.20: Performance of Model class (3.c) fit to subsets of data conditioned on k.

3.4 Unified canalization and criticality in Systems Biology Models

We analyzed the *Cell Collective* repository of biochemical models in § 2.3. Here, we analyze the same models in the light of unified canalization with regards to its ability to predict dynamical regime in comparison with their effective connectivity properties.

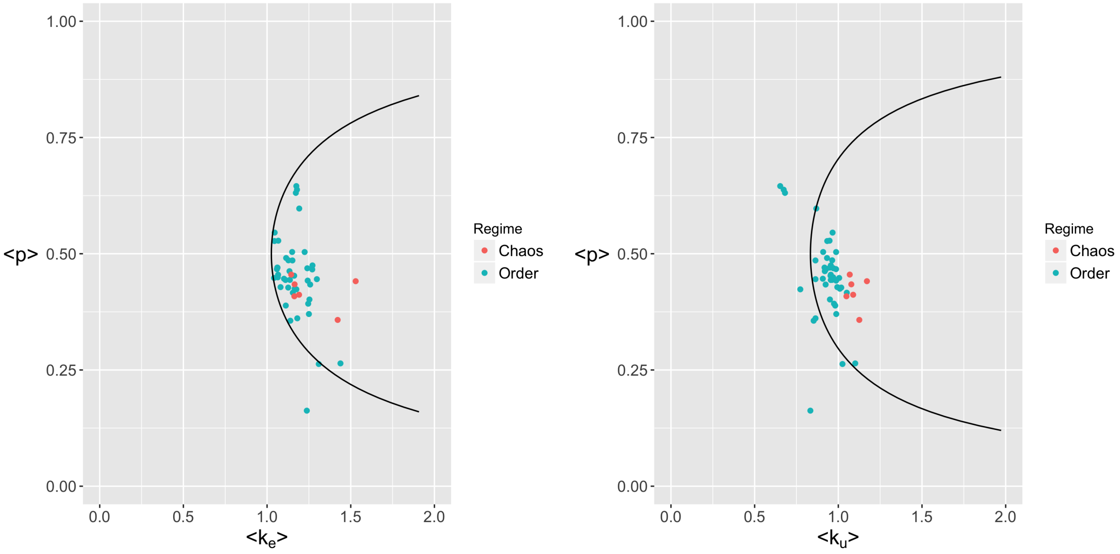
First, we observe that the mean k_u of the automata in *Cell Collective* is about 0.95, whereas the mean k_e is about 1.18 (Fig. 3.21), indicating that about 0.23 inputs contribute to input symmetry. Specifically, about 7.4% of the inputs per automaton (calculated as the mean of $(k_e - k_u)/k$) are symmetric. The proportion is even larger if we consider automata with $k > 1$ (since automata with $k = 1$ can have neither canalizing nor symmetric inputs): the mean $k_u \approx 0.89$ and the mean $k_e \approx 1.35$, indicating that about 0.46 inputs contribute to input symmetry, making up about 14.8% of the inputs per automaton.



(a) Histogram of k_e. **(b)** Histogram of k_u.

Figure 3.21: Histograms of effective connectivity (k_e) and unified canalization (k_u) of all automata nodes in the BN models of *Cell Collective*. The total number of automata considered is 2489. The vertical red line in each figure represents the mean value.

The dynamic regimes of the models are projected in two different parameter spaces, as show in Fig. 3.22: (a) $\langle k_e \rangle$-$\langle p \rangle$ and (b) $\langle k_u \rangle$-$\langle p \rangle$ spaces. The criticality models shown are respectively the criticality theory based on $\langle k_e \rangle$ derived in § 2.2.2 and the new theory based on $\langle k_u \rangle$ derived empirically in § 3.3. Here, $\langle k_e \rangle$, $\langle k_u \rangle$ and $\langle p \rangle$ represent the corresponding network-level mean values. We observe from the performances of the criticality models on the *Cell Collective* BNs, shown in Table. 3.7, that $\langle k_u \rangle$ is a better predictor than $\langle k_e \rangle$. In particular, the MCC of the criticality model utilizing $\langle k_u \rangle$ indicates an improvement of about 143% over the $\langle k_e \rangle$, even though both MCC scores are close to 0. Likewise, the R^2 is poor for both The AUC score, however, is striking: about 0.99 for the $\langle k_u \rangle$ version, an improvement of about 29% over 0.77 of the $\langle k_e \rangle$ version. In summary, the ranking performance (AUC) is good but classification performance is poor for both, and the AUC of the $\langle k_u \rangle$ model in particular is particularly excellent.

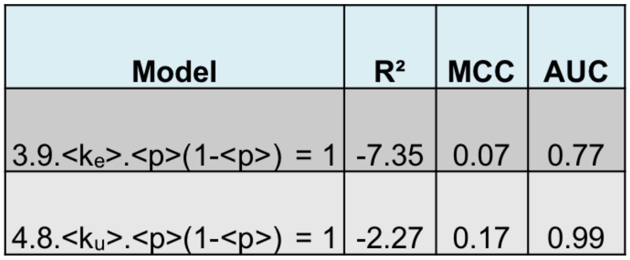


(a) $\langle k_e \rangle$-$\langle p \rangle$ space.

(b) $\langle k_u \rangle$-$\langle p \rangle$ space.

Figure 3.22: Projection of the dynamic regimes of the *Cell Collective* BNs in the phase spaces derived by fitting models to homogeneous RBNs. The parameters $\langle k_e \rangle$, $\langle k_u \rangle$ and $\langle p \rangle$ represent the corresponding network-level means of the BNs. Critical boundaries shown here are those obtained empirically in Sec. 3.3.

Note however that we have used criticality models derived from homogeneous RBNs (§ 3.3) for the

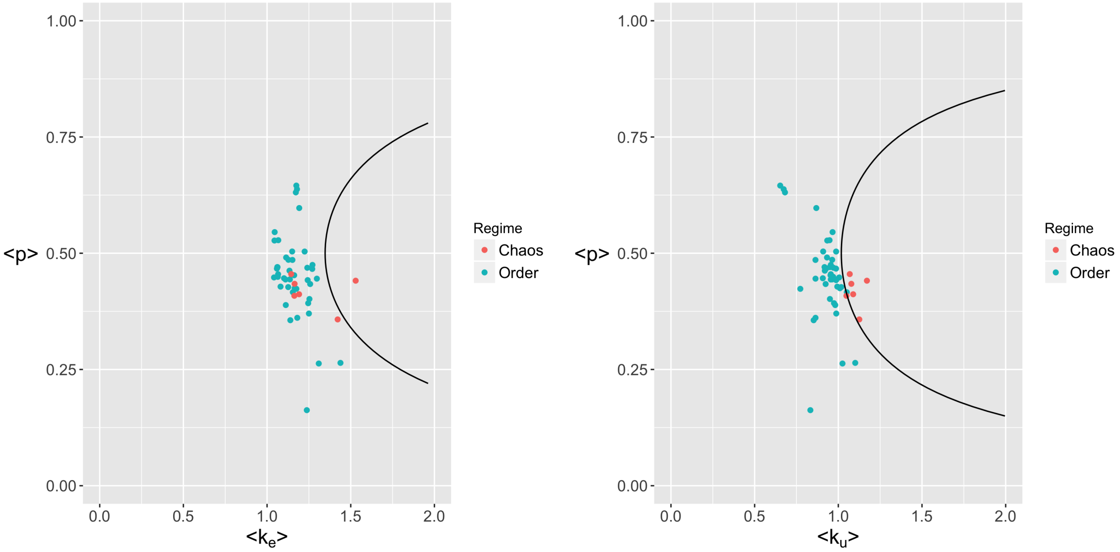


Model	R^2	MCC	AUC
3.9.<k_e>.<p>(1-<p>) = 1	-7.35	0.07	0.77
4.8.<k_u>.<p>(1-<p>) = 1	-2.27	0.17	0.99

Table 3.7: Classification performance of criticality models utilizing $\langle k_e \rangle$ and $\langle k_u \rangle$, derived in Sec. 3.3, as evaluated on the *Cell Collective* dataset.

above analysis, whereas the *Cell Collective* BNs display higher heterogeneity particularly in k and p (Fig. 2.15). There is no reason to expect that theoretical results derived for homogeneous networks would apply for heterogeneous nets such as the *Cell Collective* BNs. Therefore, we fit statistical models to only the *Cell Collective* dataset with the goal of deducing whether $\langle k_e \rangle$ or $\langle k_u \rangle$ is more *explanatory*, as opposed to being predictive, of the observed dynamical regimes. The new phase spaces thus obtained are depicted in Fig. 3.23. From the performance scores, shown in Table. 3.8, we conclude that $\langle k_u \rangle$ is again better than $\langle k_e \rangle$ at explaining the observed dynamical regimes of the *Cell Collective* BNs. In particular, the MCC of the $\langle k_u \rangle$ version shows an improvement of about 108% over the $\langle k_e \rangle$ version.

Given our empirical result that $\langle k_u \rangle$ is a worse predictor than $\langle k_e \rangle$ for networks with a small k, it is surprising to observe the roles reversed for the *Cell Collective* networks where the expected mean k of a network is only about 2.3 (Fig. 2.15). The superiority of $\langle k_u \rangle$ can be clearly discerned, besides quantified by the performance scores reported above, in Fig. 3.23 where the chaotic BNs separate from the stable ones more clearly in the $\langle k_u \rangle$-$\langle p \rangle$ space. That is, the ambiguity in the dynamical regimes for some BNs with similar values of $\langle k_e \rangle$ (and $\langle k \rangle$ as indicated in Fig. 2.16) is resolved when $\langle k_u \rangle$ is considered instead. This is more clearly seen in Fig. 3.24 where the chaotic BNs tend to have slightly higher automaton-wise k_u than the stable BNs whereas they tend to have similar automaton-wise k_e as the stable ones. Since many automata have $k = 2$ (Fig. 2.11), it is possible that many of those are logical AND, OR or their complements that have high input symmetry contributing to lower $\langle k_u \rangle$ of the stable BNs.



(a) $\langle k_e \rangle$-$\langle p \rangle$ space. **(b)** $\langle k_u \rangle$-$\langle p \rangle$ space.

Figure 3.23: Projection of the dynamic regimes of the *Cell Collective* BNs in the phase spaces derived by fitting models to the *Cell Collective* data. The parameters $\langle k_e \rangle$, $\langle k_u \rangle$ and $\langle p \rangle$ represent the corresponding network-level means of the BNs. Critical boundaries shown here are those obtained empirically by fitting statistical models to the *Cell Collective* BNs.

Model	R^2	MCC	AUC
$2.97.\langle k_e \rangle.\langle p \rangle(1-\langle p \rangle) = 1$	0.17	0.39	0.77
$3.93.\langle k_u \rangle.\langle p \rangle(1-\langle p \rangle) = 1$	0.88	0.81	0.99

Table 3.8: Classification performance of criticality models utilizing $\langle k_e \rangle$ and $\langle k_u \rangle$, derived from regression and evaluation on the *Cell Collective* dataset.

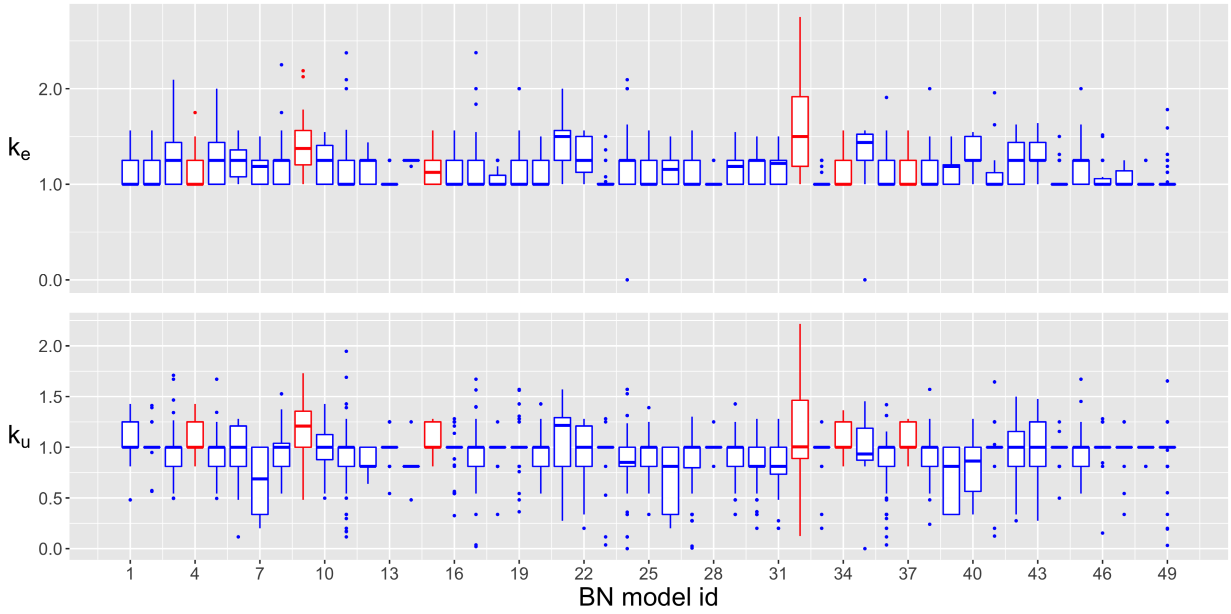


Figure 3.24: Distributions of k_e and k_u of the automata per BN model in *Cell Collective*. Boxes marked in red represent BNs in the chaotic regime and those in blue are stable nets.

Chapter 4

Integrated effective connectivity and perturbation spreading

In this chapter, we introduce 'integrated effective connectivity' (k_e^t) as an extension of effective connectivity (k_e) that incorporates the *timescale*[1] of the Boolean network's dynamics. It is defined as the effective connectivity of an *integrated* BN with timescale t which is obtained by iteratively composing a given BN with itself t times. We test the hypothesis that the mean $\langle k_e^t \rangle$ of a BN is a better predictor of dynamic regime at time $t > 1$ than the conventional mean effective connectivity $\langle k_e \rangle$. We start with the motivation for the above hypothesis, followed by the formulation of $\langle k_e^t \rangle$ and a set of experiments on a simple class of BNs known as the elementary cellular automata (ECA). Our results show that the hypothesis is true for ECA especially for larger initial perturbations spreading over longer time periods: the model with $\langle k_e^t \rangle$ as a predictor achieves a performance improvement of up to 70% in terms of MCC and up to 152% in terms of R^2 compared to the model with $\langle k_e \rangle$ for ECA. This demonstrates that integrating dynamics into a measure of effective connectivity improves prediction of dynamical regime at an arbitrary time.

Our broader goal is to close the gap between the descriptions of the BN and the dynamics at arbitrary timescales. For this reason, we further demonstrate the applicability of BN integration by using it to compare the mechanisms of two CA rules, namely 'GKL' and 'GP', designed for the

---

[1] "Timescale" can be understood as the length of the smallest unit of the scale with which we measure time (with a minimum of 1 time step)

density classification task, in order to explain their performance differences. Preliminary analysis shows that while GKL integrates smaller ranges of effective 1s (in the initial configuration of the CA) at smaller timescales into larger ranges at larger timescales, GP does it slightly the other way around.

4.1 Motivation

The phenomenon of long-term perturbation spreading in Boolean network models, and in dynamical systems in general, is of central interest, since an understanding of the same generates useful insights into the mechanics of the model, and subsequently, prescriptions for control and therapy ([3, 9, 23, 34, 70, 93, 101, 139]). Naturally, the question of what characteristics of a BN can explain long-term perturbation spreading is of importance as well, because a deeper understanding of the relationship between the BN and its long-term dynamics would enable us to better influence and control its asymptotic behavior. Recent research asks questions along these lines: how to modify the logic rules of subsets of automata in a BN so that certain attractors are stabilized [23, 139]? To answer that question, a strategy based on the short-term consequences of the combinatorial modification of logic rules was adopted, while keeping the question of the long-term consequences open [139]. Here, we explore the latter realm where we predict the behavior of a BN at any given "timescale" (formally defined below) by deducing the logic rules that dictate the dynamics at that timescale.

In the preceding chapters, we have demonstrated how to predict the dynamical regime of an RBN after one time step, as measured by the Derrida coefficient (§ 1.5.2), by using its canalization properties. That is, we bypassed the need to simulate the RBN or compute the STG to measure the dynamical regime, and instead *predict* it using the properties of the BN itself. When longer time periods are considered, can we still predict the dynamical regime in the same way? Here, we argue that the predictive power of canalization may be limited, because canalization as we measure it via k_e or k_u takes into account the effects of partial canalization of the immediate inputs, but not of inputs of inputs and so on. Considering the latter may be necessary to accurately predict long-term perturbation spreading, as the following example demonstrates.

We consider three BNs each consisting of six automata $x_1, \ldots, x_6$ and possessing the same wiring specification (Fig. 4.1). The only feature differentiating between the three cases is the logic formulas of x_4 and x_5. In all cases, nodes x_1, x_2 and x_3 are external input nodes (equivalent to automata with self-loops and COPY logic, so $k_e = 1$ and $p = 0.5$). For automata x_4, x_5 and x_6 we have $k = 2$, $p(1-p) = 0.1875$ and $k_e = 1.25$.

In each case, we compute the probability that x_6 flips its state in two time steps following a flip in x_2, denoted as $\Pr(\widetilde{x_6}(t+2)|\widetilde{x_2}(t))$ for the set of all possible initial configurations of the BN, which we observe to be different in each case (Fig. 4.1). This implies that there is no deterministic function that maps the k_e, $p(1-p)$ and the adjacency matrices of the BNs to $\Pr(\widetilde{x_6}(t+2)|\widetilde{x_2}(t))$. This is true even when p, instead of $p(1-p)$, of the automata are considered, as can be seen upon comparison of cases (a) and (b). In other words, precise prediction of behavior for individual BNs (as opposed to the average behavior of an ensemble) requires additional information than k_e and p. On the other hand, the 2-time-step logic formula for x_6 (Fig. 4.1), which specifies the logical conditions that determine the state of x_6 in two time steps and computed by combining the logic formulas of the inputs of x_6, is unique in each case. This implies that there exists a deterministic function that maps a unique characterization of these logic formulas to the respective $\Pr(\widetilde{x_6}(t+2)|\widetilde{x_2}(t))$.

Thus, the above example demonstrates the potential necessity of incorporating the combinatorial effects of the logic of the BN for the purpose of predicting perturbation spreading over longer time periods. To be sure, it may be possible to uniquely determine $\Pr(\widetilde{x_6}(t+2)|\widetilde{x_2}(t))$ in the above example by incorporating other characterizations like k_u (as it contains more information than k_e), but it is likely that such characterizations still fail to predict long-term behavior in more complicated examples. This is because it is in general not possible to analytically predict the long-term behavior of nonlinear dynamical systems [124], unlike linear systems which can be analytically solved using the tools of linear algebra.

Boolean networks are typically nonlinear systems, since Boolean functions are nonlinear with the exception of monotone functions [28]. To predict the long-term behavior of nonlinear systems specified by ordinary differential equations (ODE), 'numerical integration' techniques involving an approximation of the ODE as a discrete *iterated map*, are typically required [124], such as:

$$\begin{aligned} \frac{dx}{dt} &= f(x) \qquad (4.1)\\ \implies x(t+\Delta t) &\approx x(t) + f'(x(t))\Delta t \\ x(t+2\Delta t) &\approx x(t+\Delta t) + f'(x(t+\Delta t))\Delta t \\ &\approx x(t) + f'(x(t))\Delta t + f'(x(t) + f'(x(t))\Delta t)\Delta t \\ &\vdots \end{aligned}$$

where Δt is a sufficiently small value[2]. That is, the value of $x(t)$ at an arbitrary time t can be estimated in this way from the initial condition $x(0)$[3]. In the following section, we describe an analogous method to integrate a Boolean network and characterize the canalization of the integrated Boolean network with effective connectivity.

Thus, one of our goals is to predict dynamical regime at any given time t based on the canalization properties of the "integrated" BN which is defined by logic rules that compute the state following t time steps, rather than the canalization of the original BN which is defined by logic rules that

---

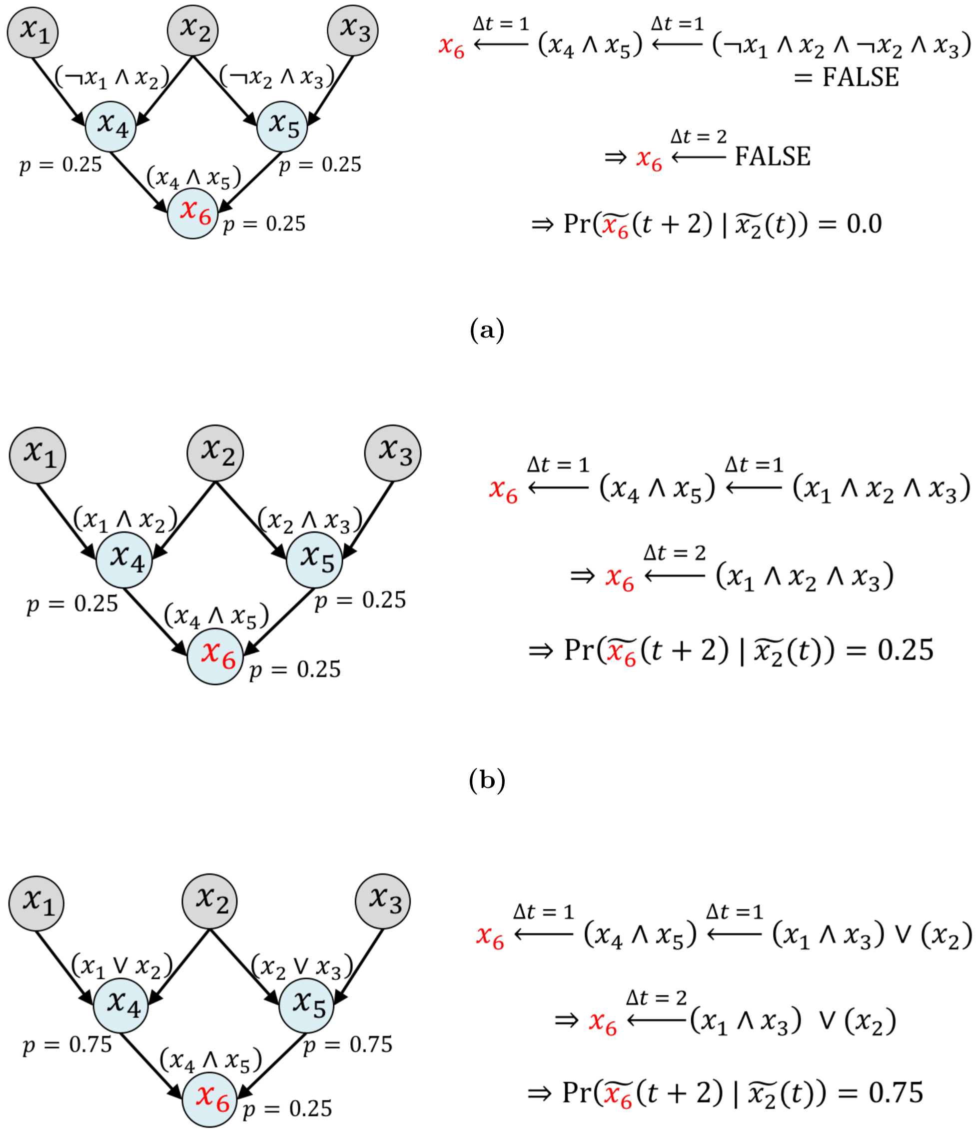
[2] Here, approximation is due to the fact that only the first derivative and a discrete value of Δt are considered.

[3] The analog of 'integration' in linear systems is just iterated scalar or matrix multiplication whose asymptotics can be precisely computed without any resort to approximation.

compute the state following one time step. The more important point is that we focus on computing a BN, rather than alternatives like an "integrated" STG that describes t-step transitions, to predict dynamic behavior at t, since it is the BN that is the ultimate target of control. As a natural consequence, a sequence of integrated BNs can yield insights into the mechanism of the BN as it evolves across timescales.

In total, our claims for BN integration are two-fold, which we support with experiments in the rest of this chapter:

1. Integration yields BNs that are more informative and thus more predictive of the dynamical regimes at the corresponding timescales.
2. Integration elucidates the mechanism of a BN.



(c)

Figure 4.1: The k_e and $p(1-p)$ of the automata, and the connectivity diagram (adjacency matrix) alone are not sufficient to predict long-term perturbation spreading. Three different BNs are shown here, with the only distinguishing feature being the logic formulas (displayed above each node) of nodes x_4 and x_5. In all three cases, the grey nodes x_1, x_2 and x_3 are external inputs; and, $k_e = 1.25$ and $p(1-p) = 0.1875$ for automata x_4, x_5 and x_6. The Δt listed above each logic formula arrow refers to the number of time steps taken by the condition specified on the right of the arrow to determine the truth of the condition specified on the left.

4.2 Boolean network integration

A Boolean network consisting of n automata $x_1, \ldots, x_n$ is defined as follows:

$$\begin{aligned} x_1^{t+1} &= F'_{1(1)}(x_1^t, \ldots, x_n^t) \\ &\vdots \\ x_n^{t+1} &= F'_{1(n)}(x_1^t, \ldots, x_n^t) \end{aligned} \tag{4.2}$$

where, x_i^t represents the state of automaton x_i at time t, and $F'_{1(i)}$ denotes the logic formula in DNF form (§ 1.2) of x_i. Here, the notation $F'_{1(i)}$, used to denote the set of wildcard schemata corresponding to the state of 1, also denotes the conventional logic formula, as the two are equivalent (§ 1.6.2).

The above system of equations constitutes an iterated map from which the states of the automata at time $t + 2$ can be computed akin to Eqn. 4.1 as follows:

$$\begin{aligned} x_1^{t+2} &= F'_{1(1)}\big(F'_{1(1)}(x_1^t, \ldots, x_n^t), \ldots, F'_{1(n)}(x_1^t, \ldots, x_n^t)\big) \\ &\vdots \\ x_n^{t+2} &= F'_{1(n)}\big(F'_{1(1)}(x_1^t, \ldots, x_n^t), \ldots, F'_{1(n)}(x_1^t, \ldots, x_n^t)\big) \end{aligned} \tag{4.3}$$

where, $F'_{1(i)}(F'_{1(1)}(.), \ldots, F'_{1(n)}(.))$ is a logic formula but not described in the standard DNF form.

The above system of equations above can be transformed into one consisting of standard DNF logic formulas:

$$x_1^{t+2} = F'^{(2)}_{1(1)}(x_1^t, \ldots, x_n^t) \tag{4.4}$$

$$\vdots$$

$$x_n^{t+2} = F'^{(2)}_{1(n)}(x_1^t, \dots, x_n^t)$$

where, $F'^{(2)}_{1(i)}$ is the standard DNF logic formula that computes the state of x_i at time step $t+2$ given the configuration $(x_1^t, \dots, x_n^t)$ at time t.

The goal of *Boolean network integration* is to compute $F'^{(\tau)} \equiv F'^{(\tau)}_0 \cup F'^{(\tau)}_1$, where $F'^{(\tau)}_0 = \{F'^{(\tau)}_{0(1)}, \dots, F'^{(\tau)}_{0(n)}\}$ and $F'^{(\tau)}_1 = \{F'^{(\tau)}_{1(1)}, \dots, F'^{(\tau)}_{1(n)}\}$, and:

$$x_1^{t+\tau} = F'^{(\tau)}_{1(1)}(x_1^t, \dots, x_n^t) \tag{4.5}$$

$$\vdots$$

$$x_n^{t+\tau} = F'^{(\tau)}_{1(n)}(x_1^t, \dots, x_n^t)$$

where, $F'^{(\tau)}_{1(i)}$ denotes the DNF logic formula that computes the state of x_i at time step $(t+\tau)$ given the configuration $(x_1^t, \dots, x_n^t)$ at time step t. The set $F'^{(\tau)}_{0(i)}$ (not shown) denotes the corresponding complementary logic formula that computes the state of $\neg x_i$.

As the time parameters of the iterated map above suggests, the integrated BN $F'^{(\tau)}_1$ computes the configuration at time step $(t+\tau)$, given any configuration at t in just one iteration which the original BN $F'^{(1)}_1$ takes τ iterations to compute[4]. For this reason, we refer to $F'^{(\tau)}_1$ as a BN with *timescale* τ[5]; the original BN $F'^{(1)}_1$ has an implicit timescale of $\tau = 1$. Furthermore, since $F'^{(\tau)}_1$ is still a BN, there is an associated state transition graph STG where a single transition represents τ consecutive transitions in the original STG associated with $F'^{(1)}_1$ (Fig. 4.2 shows an example)[6]. For example, a timescale of $\tau = 2$ means that we consider states separated by two transitions in the original STG as a single state transition.

---

[4] The integrated BN is thus equivalent to the original BN but with a τ-fold increase in speed.

[5] In particular, a *renormalized* timescale.

[6] Thus, τ can be thought of as the scale of time with which we choose to observe the dynamics of a BN.

As an important consequence of integration, the BN with timescale $\tau > 1$ contains complete information about the dynamics of the original BN that occur in exactly τ time steps. The original BN as such lacks that information since it requires additional operations (the τ update iterations) to generate that information. In other words, any characterization of the integrated BN with timescale $\tau > 1$ would be expected to more characteristic of long-term dynamics (over τ time steps) than the original BN with timescale $\tau = 1$—this point is at the core of the motivation for this chapter.

Moreover, given an integrated BN with the longest possible timescale τ^* (that computes only attractor configurations as transitions), we can compute the set of all initial configurations associated with a particular attractor. Thus, BN integration is a more general method than the "dynamics canalization" approach used to compute the same in [88].

4.2.1 Algorithm

Here, we describe a method to compute $F'^{(\tau)}$.

First, we summarize key concepts regarding the interpretation of, and operations that can be performed on, wildcard schemata sets, as pertinent to the algorithm described subsequently:

1. Each wildcard schema in $F_b'^{(\tau)}$, where $b \in \{0, 1\}$, represents a conjunctive clause, and the set $F_b'^{(\tau)}$ as a whole represents a disjunctive clause (§ 1.2). For example, in a BN with automata $x_1, \ldots, x4$, the set $F_{1(4)}'^{(1)} = \{(0, \#, 1), (1, 0, \#)\}$ associated with automaton x_4 receiving inputs from x_1, x_2, x_3 can be interpreted as: $x_4(t+1) = (\neg x_1(t) \wedge x_3(t)) \vee (x_1(t) \wedge \neg x_2(t))$.
2. Two sets of wildcard schemata can be combined by logical conjunction, following the distributive law of Boolean algebra [28], to produce a single set. For example, $\{(1, \#, \#)\} \wedge \{(\#, 1, 1), (1, 1, \#)\} = \{(1, 1, 1), (1, 1, \#)\}$. Naturally, any number of sets of wildcard schemata can be combined in this way by iteratively combining pairs of sets.

3. Two sets of wildcard schemata can be combined by logical disjunction to produce a single set. Here, disjunction is a trivial operation that simply concatenates the two sets. For example $\{(1, \#, \#)\} \vee \{(\#, 1, 1), (1, 1, \#)\} = \{(1, \#, \#), (\#, 1, 1), (1, 1, \#)\}$. Naturally, any number of sets of wildcard schemata can be combined in this way simply by concatenating all the sets.

4. A given set of wildcard schemata can be compressed in the same way as a set of LUT entries can be compressed using the Quine-McCluskey logic minimization method (§ 1.6.2). For example, the resultant schemata set in the example in point (2) above can be compressed: $\{(1, 1, 1), (1, 1, \#)\} = \{(1, 1, \#)\}$, where the first schema is absorbed by the second. Likewise, the resultant set of schemata of point (3) above can be compressed:
$\{(1, \#, \#), (\#, 1, 1), (1, 1, \#)\} = \{(1, \#, \#), (\#, 1, 1)\}$, where the third schema is absorbed by the first.

To compute the BN with the target timescale $\tau^* \geq 2$, repeat the following steps $(\tau^* - 1)$ times, starting with timescale $\tau = 2$ and moving up in increments of $\Delta\tau = 1$, for each automaton state $l \in \{0, 1\}$ and for each automaton $x_i \in \{x_1, \dots, x_n\}$:

1. Initialize an empty set A, then repeat the following for each wildcard schema in $F'^{(\tau-1)}_{l(i)}$. In a given wildcard schema, replace every literal $l_j \in \{0, 1\}$ corresponding to input x_j with the set $F'^{(1)}_{l_j(j)}$, then combine those sets by logical conjunction, and finally, compress the resulting set and append it to A.

2. Combine the sets in A by logical disjunction to obtain set B.

3. Compress the set B to obtain $F'^{(\tau)}_{l(i)}$.

The crux of the algorithm is that the BN with timescale $\tau > 1$ is obtained by combining the BN with timescale $\tau - 1$ and the original BN with timescale 1, formally represented as a *composition* operation: $F'^{(\tau)} = F'^{(1)} \circ F'^{(\tau-1)}$.

The composition operation can be better understood with the help of the following example. Consider a single automaton x_i. The set $F'^{(1)}_{1(i)}$ is a redescription of the set of all configurations at t that transition to a configuration at $t+1$ where the state of x_i is 1. The composition operation computes the set of all pre-configurations $F'^{(2)}_{1(i)}$ at $t-1$ that transition to some configuration in $F'^{(1)}_{1(i)}$, and so on. A method to compute the 'pre-image' of invidual configurations exists (see Ref. [138]), but the above algorithm is more general, in that it computes sets of pre-images of *sets* of configurations via their schema redescriptions.

The run time of the algorithm rises exponentially with the $\langle k_e \rangle$ of the BN and the target timescale τ^*. This is because: (i) the number of wildcard schemata in an automaton with k_e is approximately in the order of 2^{k_e} (§ 2.1); and (ii) for each wildcard schema, just one iteration of the algorithm requires approximately $2^{{k_e}^{k_e}}$ wildcard schemata to be combined by logical conjunction in Step 1. The effects of this combinatorial explosion can be mitigated with the help of pruning techniques (Appendix. B).

We refer to the sequences $\big(F'^{(1)}_{0(i)}, F'^{(2)}_{0(i)}, \ldots\big)$ associated with state $\neg x_i$ and $\big(F'^{(1)}_{1(i)}, F'^{(2)}_{1(i)}, \ldots\big)$ associated with state x_i as the *Boolean function sequence* (BFS), and the sequence $\big(\{F'^{(1)}_{(1)}, \ldots, F'^{(1)}_{(n)}\}, \{F'^{(2)}_{(1)}, \ldots, F'^{(2)}_{(n)}\}, \ldots\big)$ as the *Boolean network sequence* (BNS)[7].

A detailed example is described below. More details of the integration algorithm are available in Appendix. B. We emphasize that the above algorithm is not meant to compute the STG which can be computed from the original BN F' itself. Although, every $F'^{(\tau)}$ generated by the algorithm has an associated STG, as described above. Just as the relationship between a conventional BN and the associated STG, $F'^{(\tau)}$ is a *compressed* description, a model, of the associated STG that describes the dynamical transitions occurring at timescale τ.

---

[7] Such sequences are formally known as 'Picard sequences' [133].

4.2.2 Example

Here, we fully describe the BFS and BNS of a small Boolean network.

Fig. 4.2 depicts a BN with three automata and the associated STG. The associated BFS sequences are shown in Fig. 4.3, which reach a period-3 *attractor*[8] starting at $\tau = 2$; both BFS sequences show this attractor since they are just logical complements of each other.

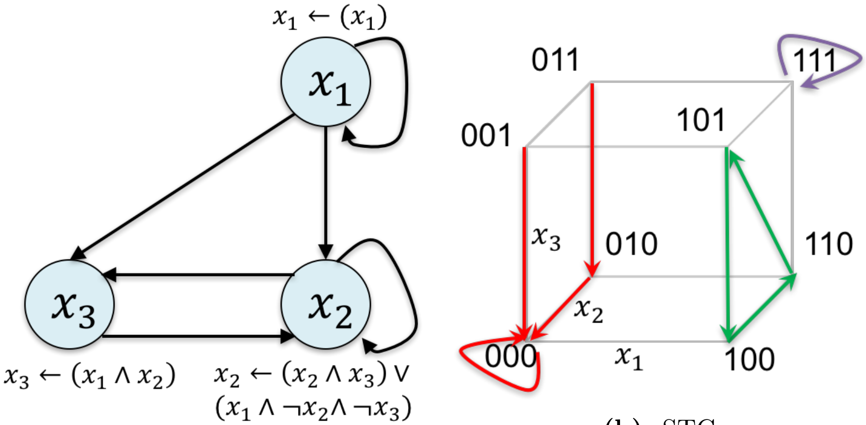
The BFS attractor naturally corresponds to a BNS attractor, as shown in the BNS sequence shown in Fig. 4.4. This is because a BN with timescale τ is simply a collection of the wildcard schemata associated with every automaton, shown in the corresponding column of Fig. 4.3(b).

We observe from Fig. 4.4 that interactions among the automata could be altered at longer timescales. Consider x_2 and x_3 for instance. At $\tau = 1$, they influence each other, whereas at $\tau = 3$ they don't interact, as seen in the corresponding logic formulas and even in just their interactions. This means that a perturbation caused in the state of x_2 can propagate to x_3 in one time step, but not at exactly in three time steps. This can be more clearly seen in the corresponding STGs (bottom of Fig. 4.4). Take for instance, the configuration $(1, 0, 1)$ and the configuration $(1, 1, 1)$ caused by flipping the state of x_2 in the first. These two configurations respectively transition to $(1, 0, 0)$ and $(1, 1, 1)$ in one time step, thus propagating the perturbation to x_3; whereas, in three steps they return to their original configurations $(1, 0, 1)$ and $(1, 1, 1)$ respectively, preserving the original perturbation in x_2 but not propagating it to x_3.

The BFS can be viewed as a decomposition of the STG, and there are several properties (for example, the properties of the transients and the attractors in the BFS and STG are clearly related) that merit a dedicated study in the future.

---

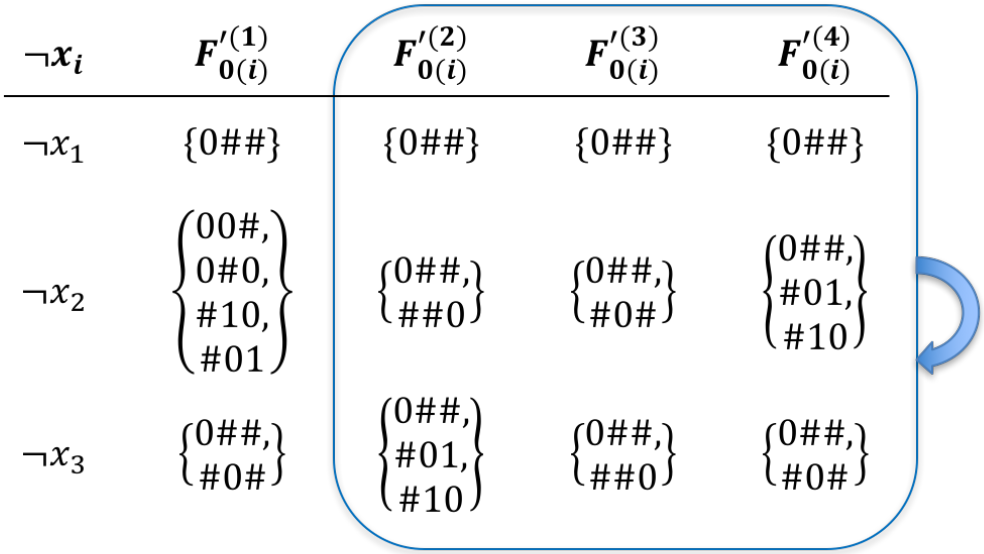
[8] Formally known as the solution of the iterated map of Eqn. 4.4 [25, 74, 100].



(a) Example BN

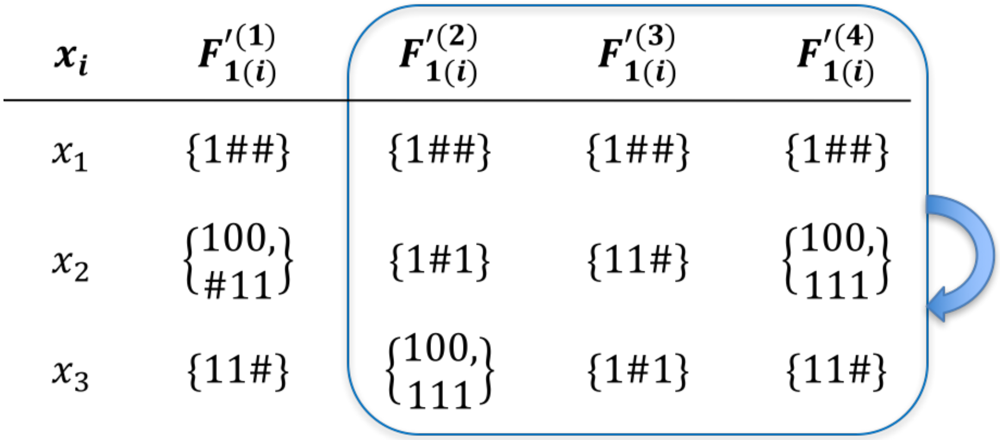
(b) STG

Figure 4.2: An example BN and its state transition graph (STG) depicted in Euclidean space. The logic formulas are listed next to each node, and the colors in the STG correspond to the basins of attraction.



$\neg x_i$	$F'^{(1)}_{0(i)}$	$F'^{(2)}_{0(i)}$	$F'^{(3)}_{0(i)}$	$F'^{(4)}_{0(i)}$
$\neg x_1$	{0##}	{0##}	{0##}	{0##}
$\neg x_2$	{00#, 0#0, #10, #01}	{0##, ##0}	{0##, #0#}	{0##, #01, #10}
$\neg x_3$	{0##, #0#}	{0##, #01, #10}	{0##, ##0}	{0##, #0#}

(a) BFS associated with state $\neg x_i$.



x_i	$F'^{(1)}_{1(i)}$	$F'^{(2)}_{1(i)}$	$F'^{(3)}_{1(i)}$	$F'^{(4)}_{1(i)}$
x_1	{1##}	{1##}	{1##}	{1##}
x_2	{100, #11}	{1#1}	{11#}	{100, 111}
x_3	{11#}	{100, 111}	{1#1}	{11#}

(b) BFS associated with state x_i.

Figure 4.3: The BFS sequences of the three automata of the BN shown in Fig. 4.2(a). The blue box highlights the 'attractor' of the BFS—a block of the sequences that repeats itself after its first occurence (indicated by the thick blue arrow). Here, we see a period-3 attractor starting at $\tau = 2$, and repeats itself at $\tau = 5, 8, \ldots$. In order to avoid keeping track of the input variables corresponding to the literals in each schema separately, we fill every schema with wildcard symbols at appropriate locations so that all schemata have exactly $n = 3$ symbols. Thus, a symbol in the i^{th} position of a schema refers to the state (0, 1, or #) of automaton x_i.

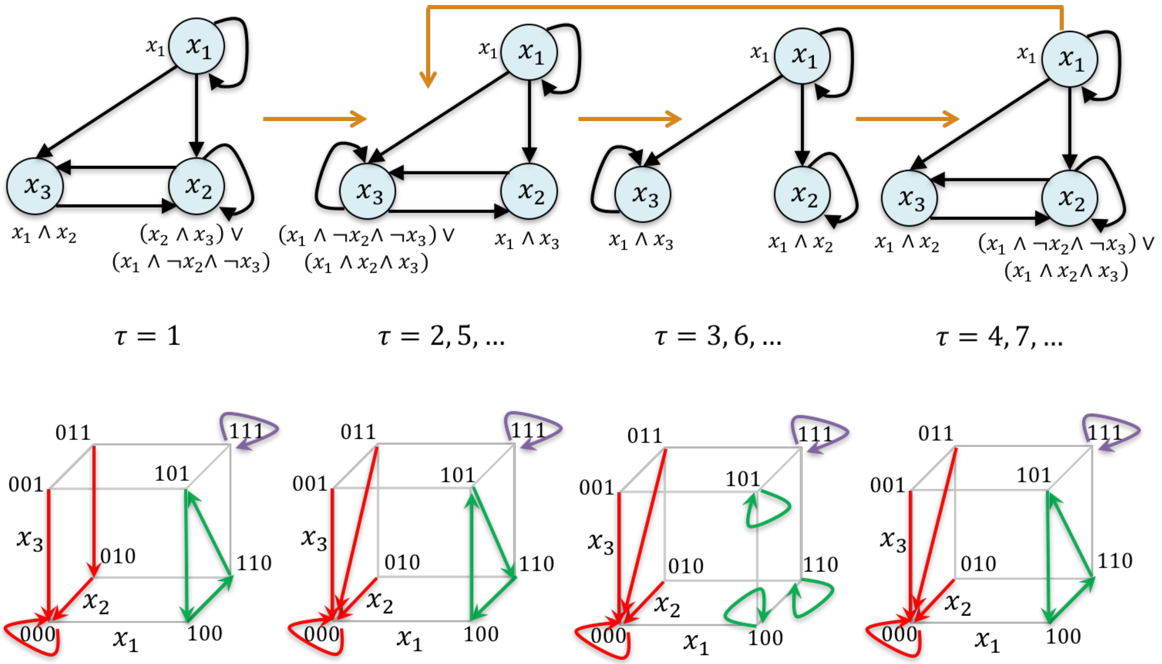


Figure 4.4: The BNS of the BN shown in Fig. 4.2(a) and the corresponding STGs. (Top) The BNS sequence, with the associated timescales τ (middle) and the STGs (bottom). The left-most BN and the STG are the same as in Fig. 4.2. The logic formulas are equivalent to the wildcard schemata in Fig. 4.3(b) (see main text for explanation). The STG associated with a BN with timescale τ can also be deduced from the STG corresponding to $\tau = 1$ by replacing every sequence of τ consecutive arrows with just one. Every transition in an STG with timescale τ can be computed from the BN with timescale τ in just one iteration.

4.3 Integrated effective connectivity and output bias

We can characterize the canalization of every BN in the BNS $\big(\{F'^{(1)}_{(1)},\ldots,F'^{(1)}_{(n)}\},\{F'^{(2)}_{(1)},\ldots,F'^{(2)}_{(n)}\},\ldots\big)$ with measures such as $\langle k_e\rangle$ and $\langle k_u\rangle$ in the same way as we have previously characterized the canalization of the first BN of the sequence. Here, we introduce *integrated effective connectivity* (k_e^t) as a measure of effective connectivity of an automaton in an integrated BN $F'^{(\tau)}$; when $\tau = 1$, $k_e^1 = k_e$.

The k_e^τ of automaton x_i in the BN $F'^{(\tau)}$ can be computed using the wildcard schemata sets $F'^{(\tau)}_{0(i)}$ and $F'^{(\tau)}_{1(i)}$, in the same way as $k_e^1 = k_e$ is computed using $F'^{(1)}_{0(i)}$ and $F'^{(1)}_{1(i)}$ (§ 1.6.3). Similarly, p^τ, representing the output bias of automaton x_i in $F'^{(\tau)}$, can be computed.

The averages of k_e^τ and p^τ over all automata of the BN $F'^{(\tau)}$ are respectively denoted as $\langle k_e^\tau\rangle$ and $\langle p^\tau\rangle$. In the experiments below, we consider $\langle p^\tau\rangle(1-\langle p^\tau\rangle)$ instead of just $\langle p^\tau\rangle$ as before.

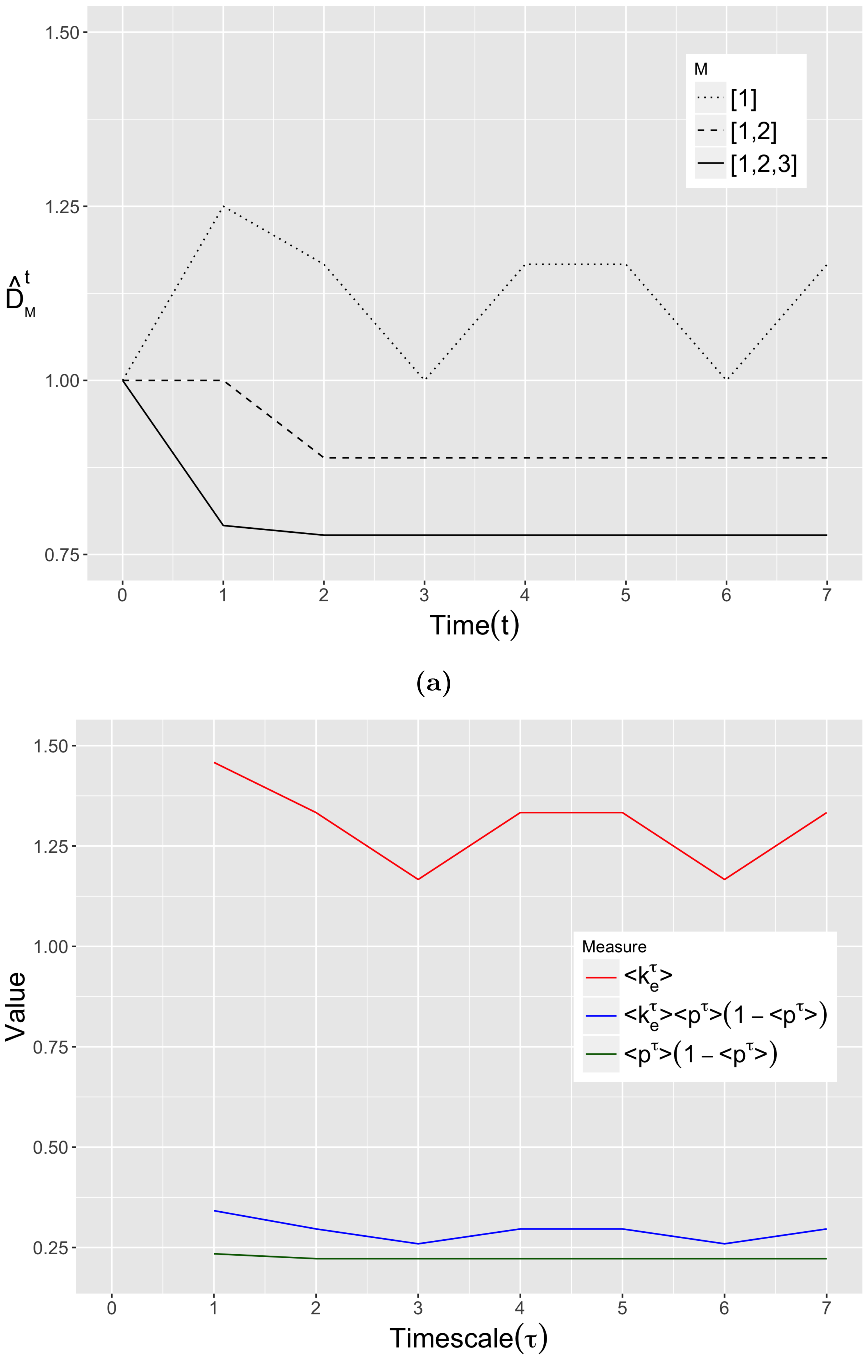
In summary, we consider the following measures of canalization and output bias in the integrated BNs:

1. $\langle k_e^\tau\rangle$
2. $\langle p^\tau\rangle(1-\langle p^\tau\rangle)$
3. $\langle k_e^\tau\rangle\langle p^\tau\rangle(1-\langle p^\tau\rangle)$

We refer to these characteristics of automata as *local* measures when $\tau = 1$, and as *integrated* measures when $\tau > 1$.

Fig. 4.5 shows the values of all three measures over the timescales $\tau = 1,\ldots,7$ for the example BN described in the preceding section. Additionally, it shows the properties of perturbation spreading, as indicated by $\widehat{D}^t_M$ (§ 1.5.2) of the same BN at time steps $t = 0,\ldots,7$ and for three different possible ranges of M. As explained in § 1.5.2, $\widehat{D}^t_M = 1$ when $t = 0$.

We observe that the behaviors of $\langle k_e^\tau \rangle$, $\langle p^\tau \rangle(1 - \langle p^\tau \rangle)$ and $\langle k_e^\tau \rangle \langle p^\tau \rangle(1 - \langle p^\tau \rangle)$ visually match the perturbation spreading behavior indicated by $\widehat{D}_M^t$. In particular, $\langle k_e^\tau \rangle$ is most characteristic of perturbation spreading for small initial perturbation sizes $M = [1, 1]$; $\langle p^\tau \rangle(1 - \langle p^\tau \rangle)$ for large $M = [1, 3]$; and, $\langle k_e^\tau \rangle \langle p^\tau \rangle(1 - \langle p^\tau \rangle)$ for the intermediate range $M = [1, 2]$. One possible explanation for this is that k_e is more of a 'local' measure, being a characterization of subsets of LUT entries that are closely in the hypercube; whereas, p is a more of a 'global' measure, being a characterization of all LUT entries going to a particular output regardless of their locations in the hypercube. As we discuss below, these relationships hold for a whole class of BNs.



(a)

(b)

Figure 4.5: Properties of the integrated BNs and perturbation spreading in the example BN of Fig. 4.2 during the interval $t = 0, \ldots, 7$. (a) Perturbation spreading for all different ranges of initial perturbation sizes M. (b) Measures of canalization and output bias of the integrated BNs at each $t \geq 1$.

4.4 Efficacy of the integrated characteristics in predicting long-term dynamical regime

We compare the local and integrated measures of a BN as predictors of dynamical regime at $\tau \geq 1$ in a small class of BNs known as the *elementary cellular automata* (ECA). Specifically, our goal is to establish whether or not the integrated measures are more predictive than the local measures for ECAs, as opposed to producing particular expressions for predicting dynamical regime for a broad class of BNs as we pursued in the previous chapters.

An ECA is a BN whose nodes are embedded on a single-dimensional lattice with periodic boundaries, that is the topology is a ring lattice, and has the following properties: $\forall i \in \{1, \ldots, n\}$, $k_i = 3$, $X_i = (x_{i-1}, x_i, x_{i+1})$ and $F_i = F_j, \forall j \in \{1, \ldots, n\}$ (see Sec. 1.3 for the definition of a BN). Consequently, the mean values of the integrated measures of an ECA are the same as the values of the individual automata. That is, $\langle k_e^\tau \rangle = k_e^\tau$, $\langle p^\tau \rangle (1 - \langle p^\tau \rangle) = p^\tau (1 - p^\tau)$ and $\langle k_e^\tau \rangle \langle p^\tau \rangle (1 - \langle p^\tau \rangle) = k_e^\tau p^\tau (1 - p^\tau)$.

Due to the aforementioned properties, an ECA can be uniquely identified by the associated *Rule number* which is the decimal equivalent of the binary output vector of the associated F [137]. Since there are 256 unique ECAs, the Rule numbers range from 0 to 255.

We chose ECAs because they are particularly useful as test beds for the integrated measures, since global connectivity—a potential factor in the prediction of long-term dynamical behavior—is not a variable (hence, not a variable in the integrated ECAs as well). In a given ECA, furthermore, since all automata have the same F, the distributions of k, k_e and p have zero variance. Thus, the differences in the long-term dynamic regimes of the ECAs can only be attributed to the the differences in the properties of F and $F'^{(\tau)}$.

We consider the full ensemble of 254 non-constant ECAs with $n = 10$ automata each. Of these, 226 of them settle on to an attractor before $t = 10$ steps, that is their maximum transient length is less than 10. The average length of the transient in the full ensemble is about 6.5 steps. For these reasons and also due to the computational complexity of the integration procedure, we consider a small time interval $t = 1, \ldots, 10$ over which we study the behavior of the ECAs. Due to the small n and a small interval of t, we computed $\widehat{D}_M^t$ by considering all $\frac{1}{2}\binom{2^n}{2}$ pairs of initial configurations.

In summary, we follow a computational approach to compare the local and integrated automaton measures as predictors of the long-term dynamical regime of the ECAs:

1. Compute k_e^τ, $p^\tau(1-p^\tau)$ and $k_e^\tau p^\tau(1-p^\tau)$ at each $\tau = 1, \ldots, 10$ of the full ensemble containing 254 ECAs.
2. Compute the dynamic regimes based on $\widehat{D}_M^t$ and the corresponding D_c^t at each $t = 1, \ldots, 10$ of the ECAs for various values of M.
3. Compare classification and regression performance of statistical models utilizing the local and integrated measures as predictors of D_c^t, at each t and M.

4.4.1 Results of Experiments

We analyze the data obtained from above to answer the following questions:

- How well do the statistical models utilizing either the local predictors, namely k_e^1, $p^1(1-p^1)$ and $k_e^1 p^1(1-p^1)$ or the integrated predictors, namely k_e^τ, $p^\tau(1-p^\tau)$ and $k_e^\tau p^\tau(1-p^\tau)$, predict the observed dynamical regime D_c^t when $\tau = t$?
- Do models utilizing the integrated predictors with timescale parameter τ perform better than the equivalent models containing the local predictors instead, in predicting D_c^t when $\tau = t$?

We formulate the above questions as a binary classification problem, and answer them using measures of classification and regression performance.

First, we make a few observations on the data. Fig. 4.6(a) shows the behavior of $\widehat{D}^t_M$, for $M = [1, 3]$, of the full ensemble. We observe that perturbations amplify in most ECAs, that is, $\widehat{D}^t_M > \widehat{D}^0_M = 1$, for $t \geq 1$, typically. Furthermore, in many ECAs perturbations oscillate as they spread. Fig. 4.6(b-d) show the behavior of the integrated measures of the ECAs, and they too display oscillations, qualitatively matching the perturbation spreading pattern.

We also observe the following about the behavior of Rule 30 known for its chaotic dynamics and has been used as a random number generator [137]. The relatively high sensitivity to perturbations of Rule 30 (blue), as seen in the quick increase in the mean perturbation size with time, is matched by a concomitant rise in the corresponding k_e^τ and $k_e^\tau p^\tau(1 - p^\tau)$. This leads to the conjecture that the latter is an explanation of the former, which will be investigated in future work. Together, these observations hint at the possibility that the integrated measures are predictive of long-term perturbation spreading behavior.

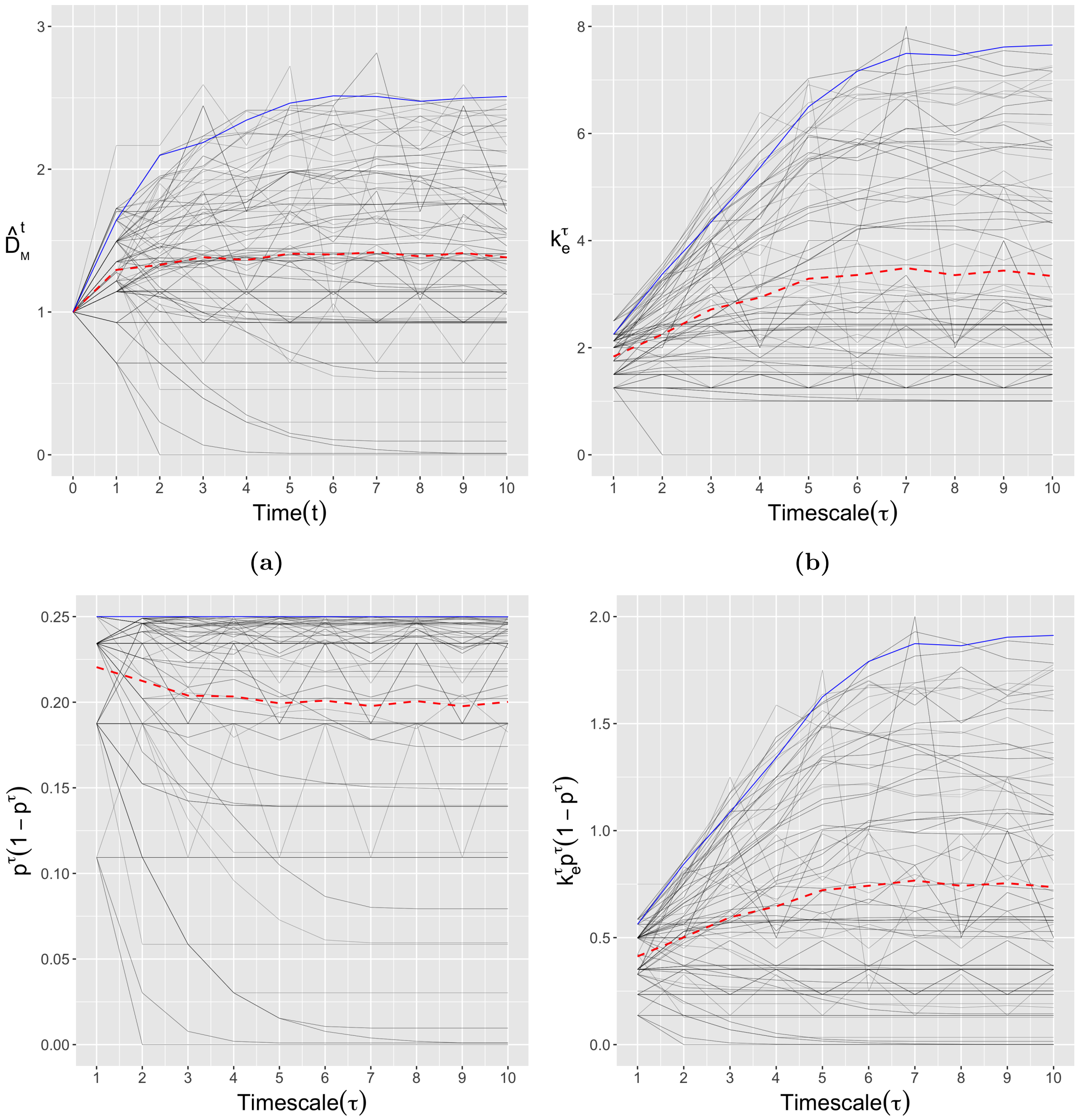
To compare the abilities of the local and the integrated measures of ECA in predicting dynamical regimes over a given time interval, we consider a set of statistical models whose general form is:

$$R^t_M = \text{step}(\text{logistic}(\text{Model}))$$

where, R^t_M is the dynamical regime at time step t given the range of initial perturbation sizes M, and the value of the step function is 1 if the output of the logistic function is greater than a threshold of 0.5, and 0 otherwise.

Each model within the logistic function belongs to one of the following "Model classes":

1. **a)** $c_1.k_e^1$; **b)** $c_1.k_e^t$



(a)

(b)

(c)

(d)

Figure 4.6: The properties of perturbation spreading and integrated automaton characteristics of the full ensemble of ECAs with all 254 non-constant rules. The range of initial perturbation sizes in (a) is $M = [1, 3]$. The dashed red line in each plot represents the average trend (shown just as a visual guide), while the solid blue line represents the behavior of Rule 30, known for its chaotic properties and used as a random number generator.

2. a) $c_1.p^1(1-p^1)$; **b)** $c_1.p^t(1-p^t)$

3. a) $c_1.k_e^1 p^1(1-p^1)$; **b)** $c_1.k_e^t p^t(1-p^t)$

where, we have set $\tau = t$ for the integrated predictors.

In summary, we have both a regression model and a binary classifier; the step function turns the logistic model into a classifier. We used R^2 to evaluate the regression performance and MCC to evaluate the classification performance (both defined in § 2.2.2). We consider only MCC for evaluating classification performance because the set of class labels tends to be unbalanced at each t, at least for smaller ranges of M, as shown in Fig. 4.7. Also, since our goal is not to produce particular predictive models, we will not evaluate their ability to generalize via cross-validation.

As before, we define 'performance difference' between models in a given Model class as follows:

$$\text{Performance difference} = \frac{\text{Performance(Model x.b) - Performance(Model x.a)}}{\text{Performance(Model x.a)}} \tag{4.6}$$

where $x \in \{1, 2, 3\}$.

Figs. 4.8, 4.9 and 4.10 depict the performances of the models in each of the three Model classes listed above, where each model is evaluated for $M = [1, 1]$, $M = [1, 4]$ and $M = [1, 7]$, and for $t = 1, \ldots, 10$. Figs. 4.11, 4.12 and 4.13 depict the corresponding performance differences.

We observe the following:

1. The integrated predictor is almost always better than the local predictor in Model class 3, in terms of both classification and regression performance. For $M = [1, 1]$ in particular, most models (3.b) show maximal performance irrespective of t. With larger M, their MCC scores drop to a minimum of about 55% at $t = 4$, and with a performance difference over the

corresponding model (3.a) of about 27%.

2. The performances vary systematically with M depending on the Model class: for Model class 1, both MCC and R^2 tend to be lower with larger M, whereas for Model class 2, they tend to be larger with larger M. In terms of R^2, Models (b) tend to perform better for larger M, irrespective of the Model class; in particular, the performance differences systematically increase with M. In terms of MCC, more Models (b) tend to be better with larger M, although the magnitude of performance difference shows no discernible pattern as M varies.
3. Models (b) tend to perform better with t, attaining a maximal performance gain over Models (a) at around $t = 5$.

A possible explanation for observation 1 and 2 is that larger initial perturbations recruit more automata, thus more interactions among them, in the spreading process. A possible explanation for observation 3 is that the perturbations tend to complete a full circle of propagation through the ECA (of 10 automata) at around $t = 5$. That is also the time by which the integration procedure would have swept through all of the automata, and thus would be expected to contain the most predictive of long-term perturbation spreading, compared to the original ECAs which receive inputs only locally.

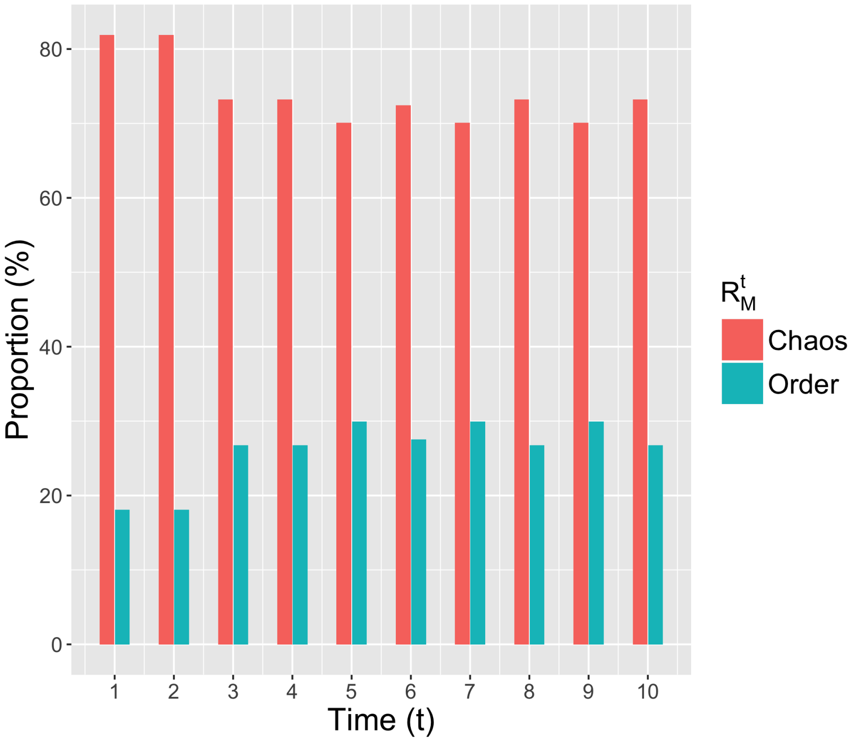
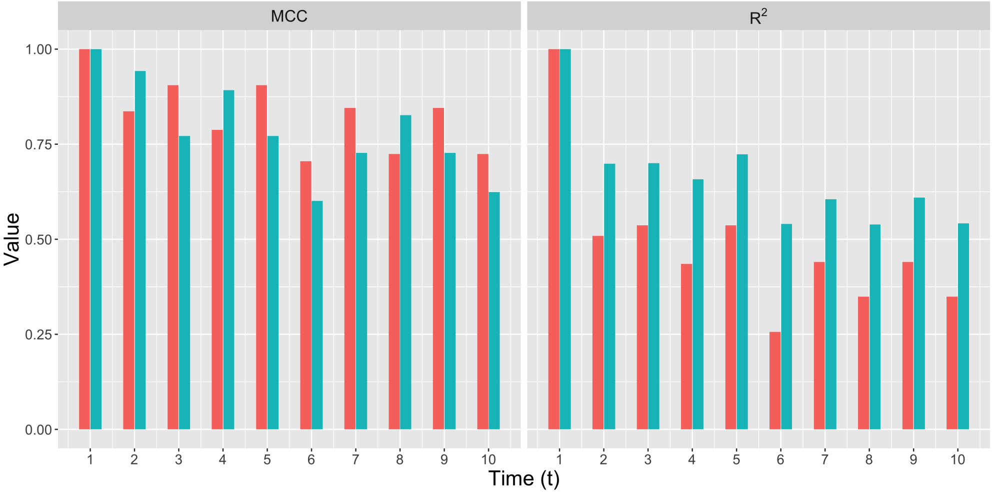
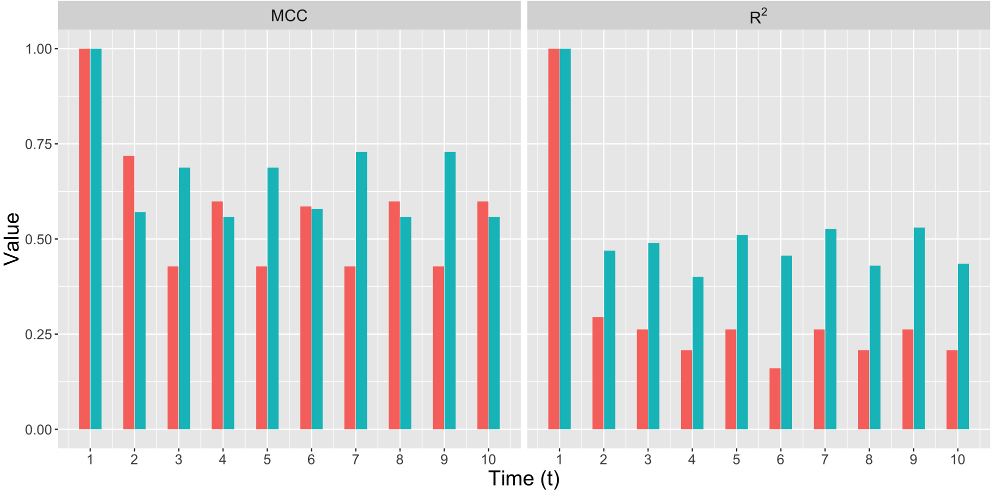


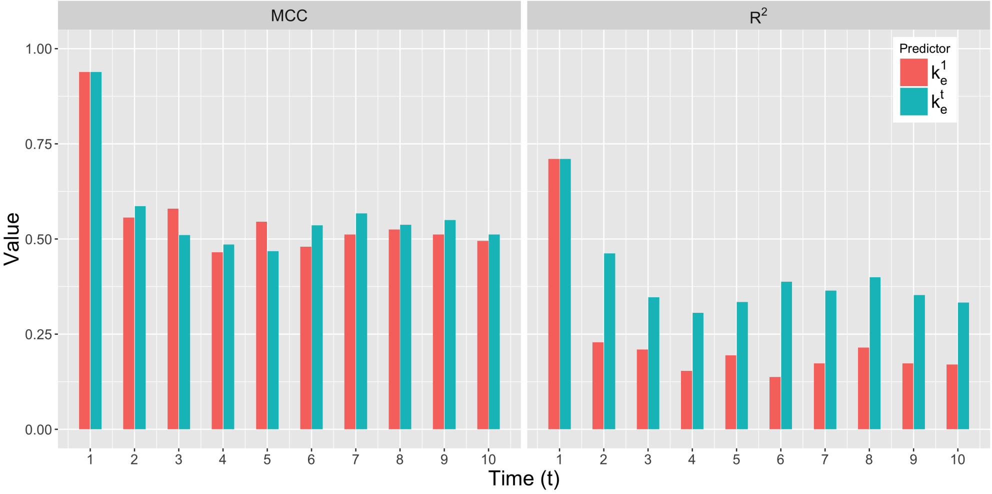
Figure 4.7: The proportions of the two different dynamical regimes at each t for $M = [1, 3]$.



(a) $M = [1, 1]$.

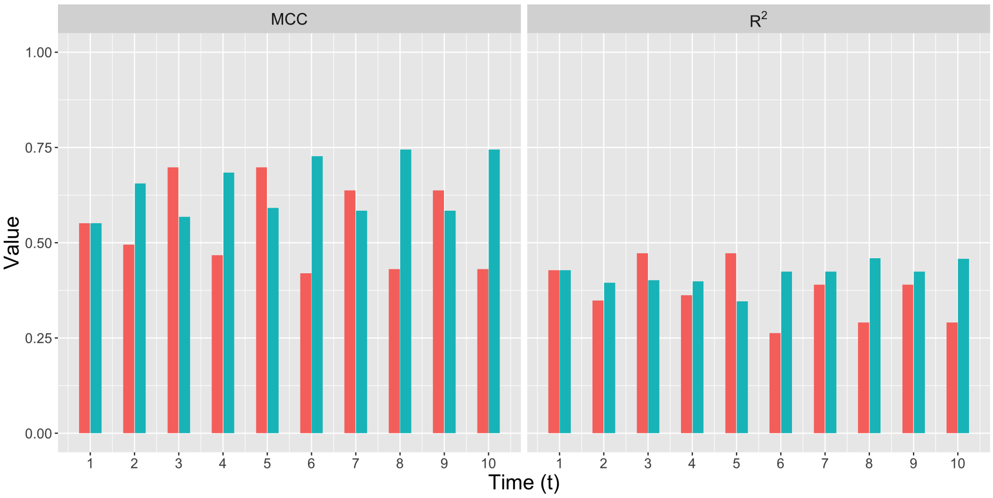


(b) $M = [1, 4]$.

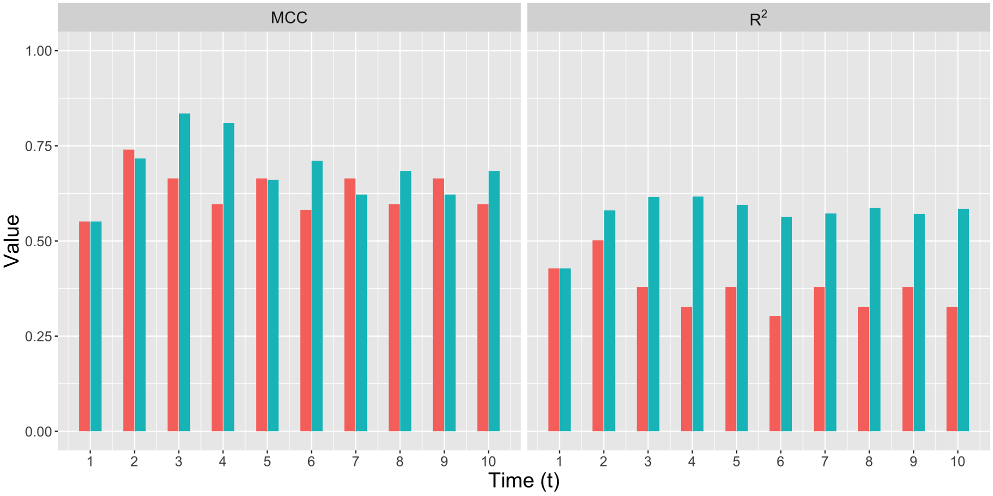


(c) $M = [1, 7]$.

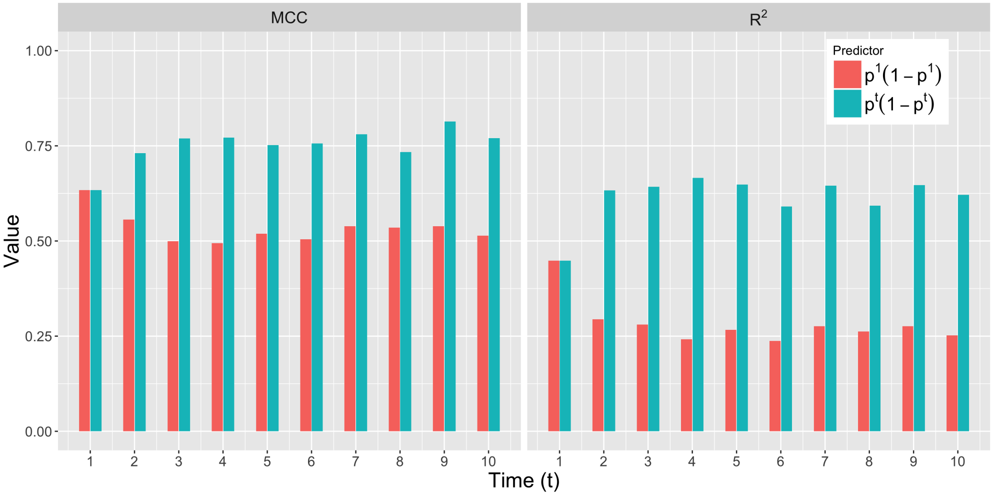
Figure 4.8: Classification and regresssion performances of Models 1.a and 1.b for different values of M.



(a) $M = [1, 1]$.

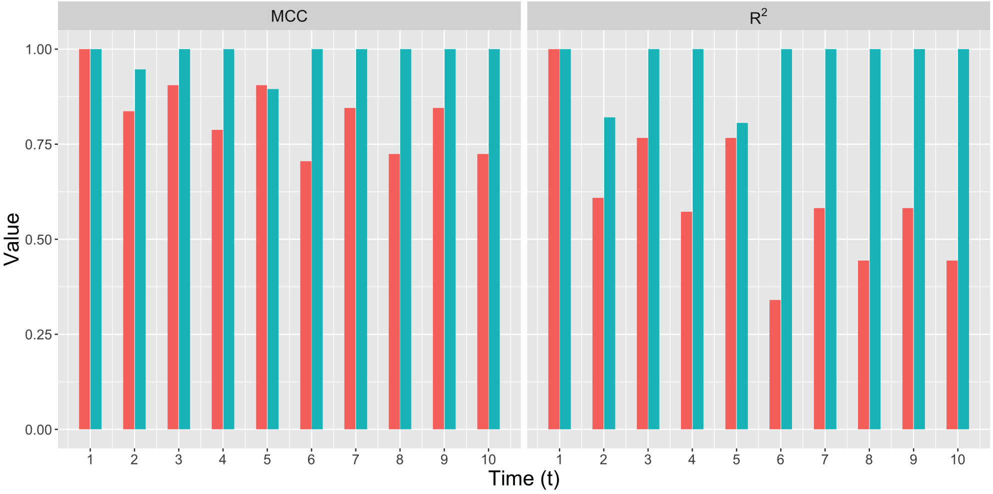


(b) $M = [1, 4]$.

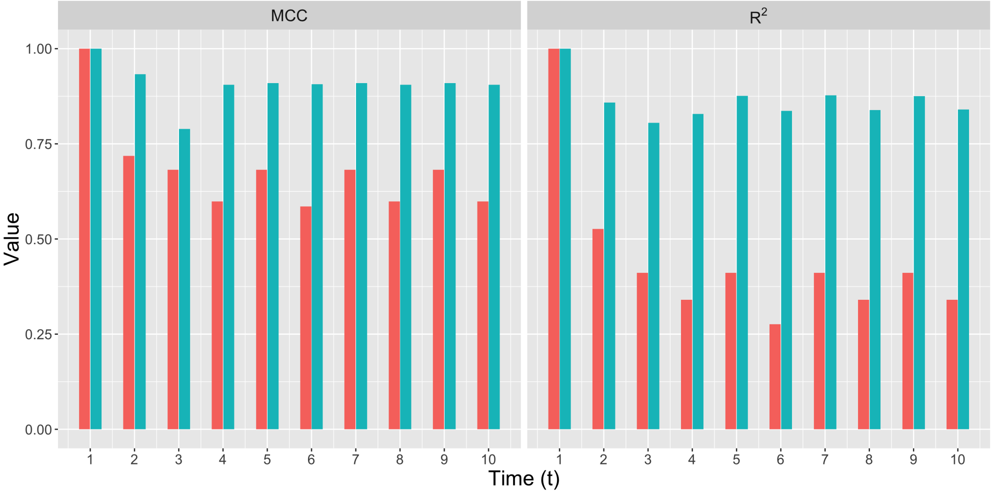


(c) $M = [1, 7]$.

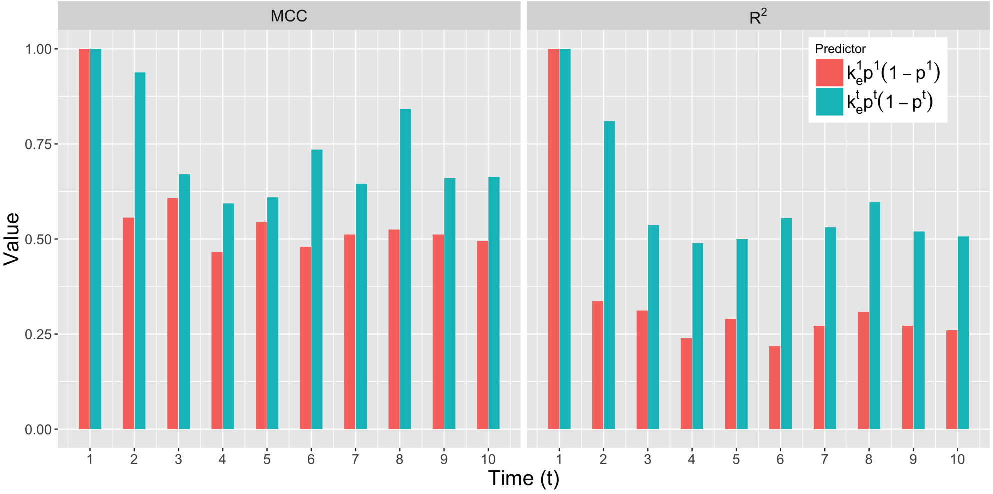
Figure 4.9: Classification and regresssion performances of Models 2.a and 2.b for different values of M.



(a) $M = [1, 1]$.

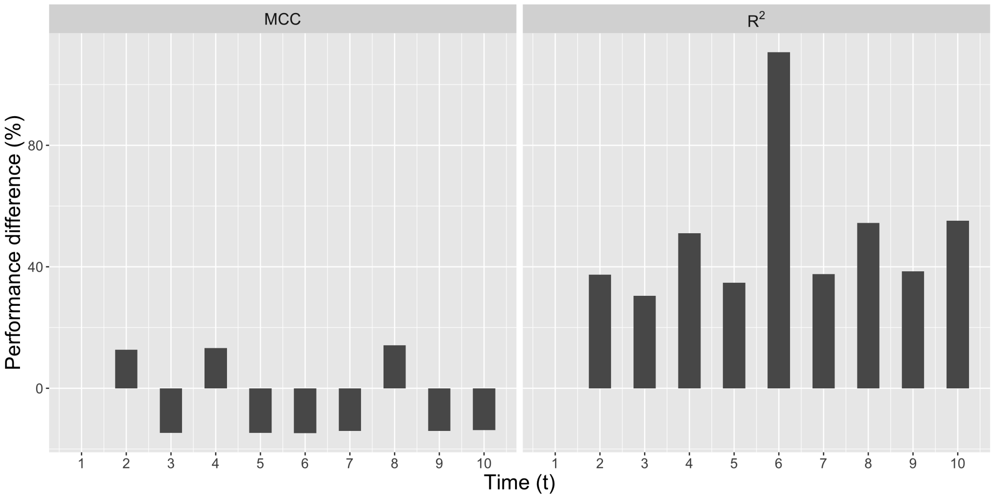


(b) $M = [1, 4]$.

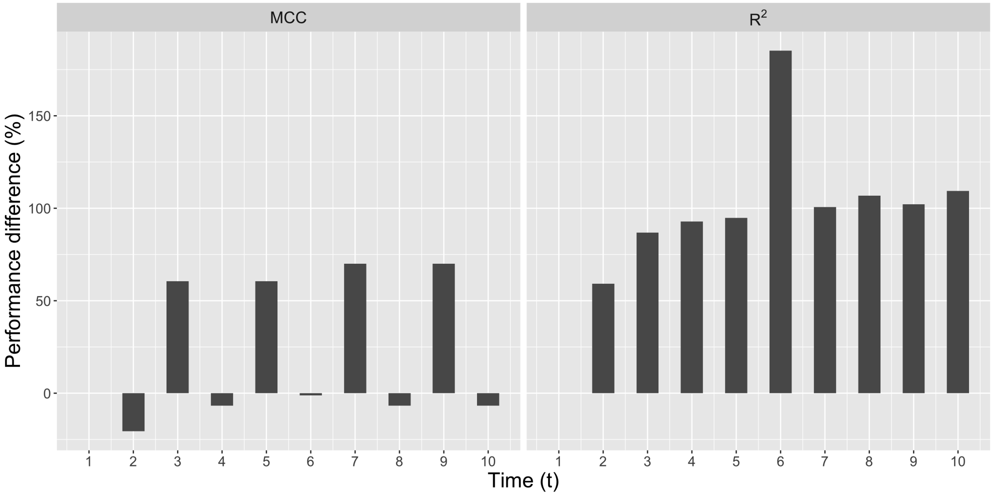


(c) $M = [1, 7]$.

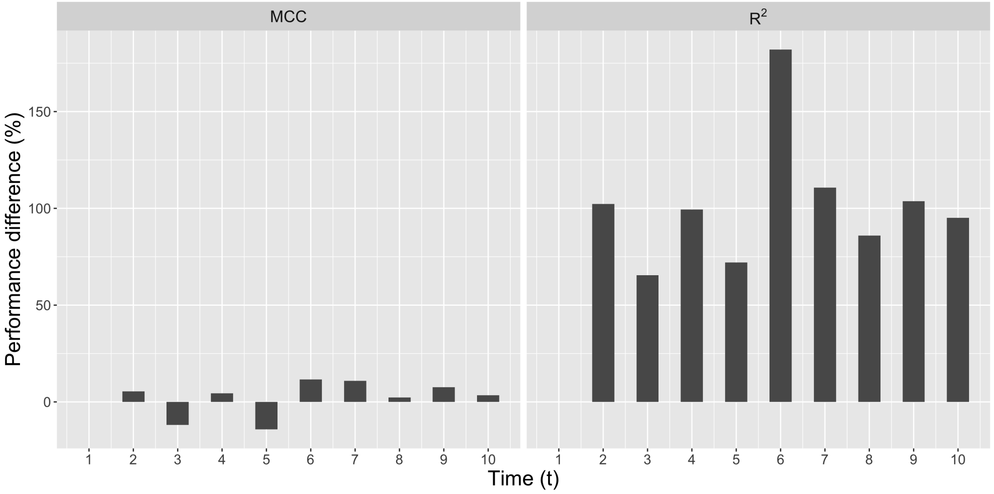
Figure 4.10: Classification and regresssion performances of Models 3.a and 3.b for different values of M.



(a) $M = [1, 1]$.

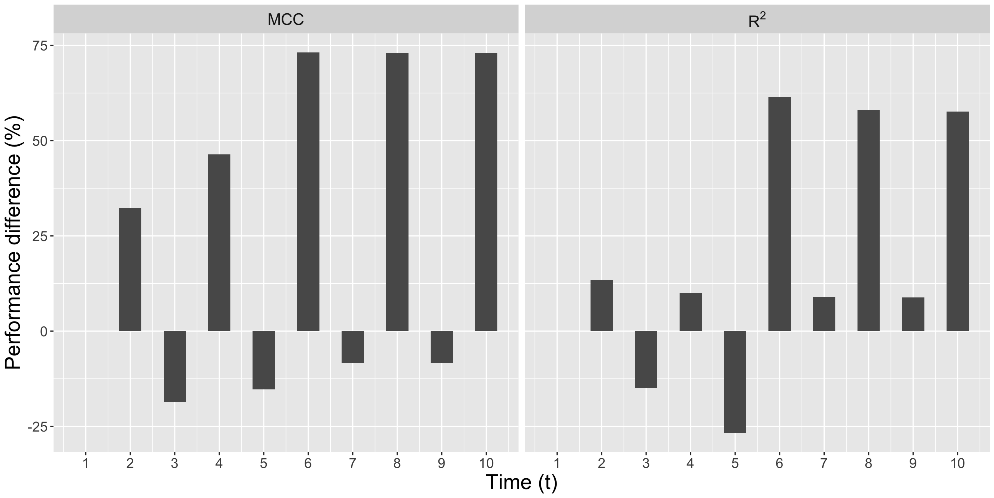


(b) $M = [1, 4]$.

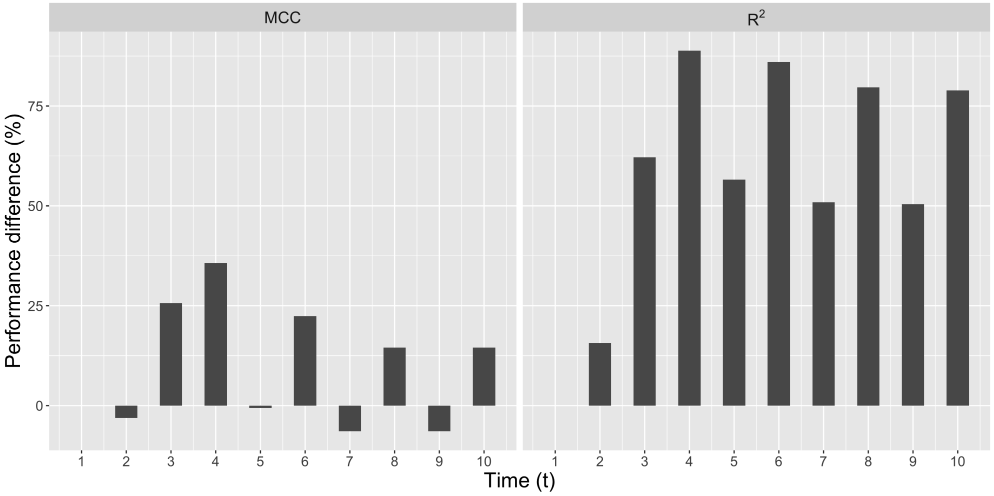


(c) $M = [1, 7]$.

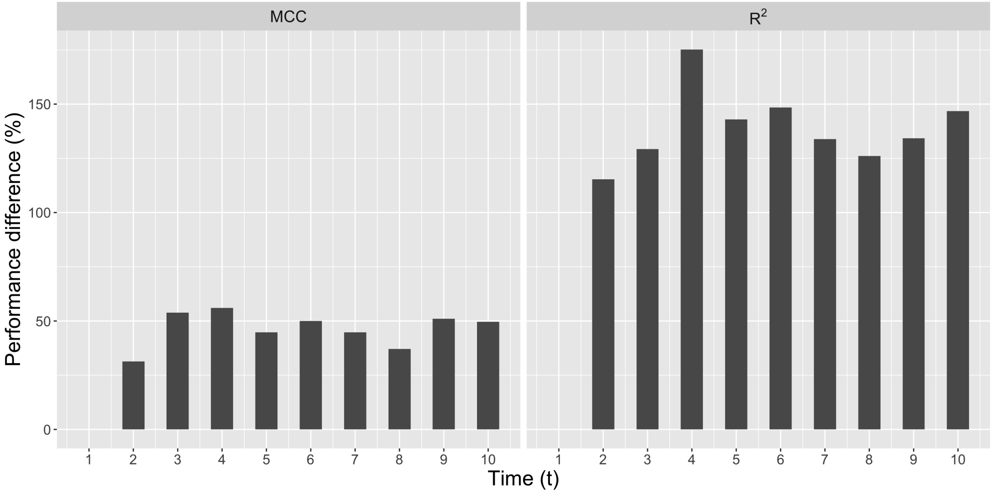
Figure 4.11: Performance differences between Models 1.a and 1.b for different values of M.



(a) $M = [1, 1]$.

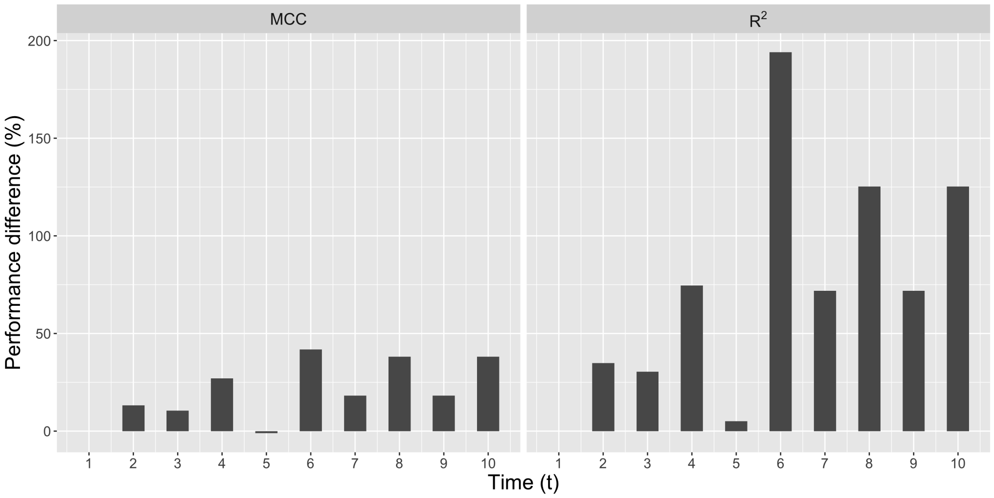


(b) $M = [1, 4]$.

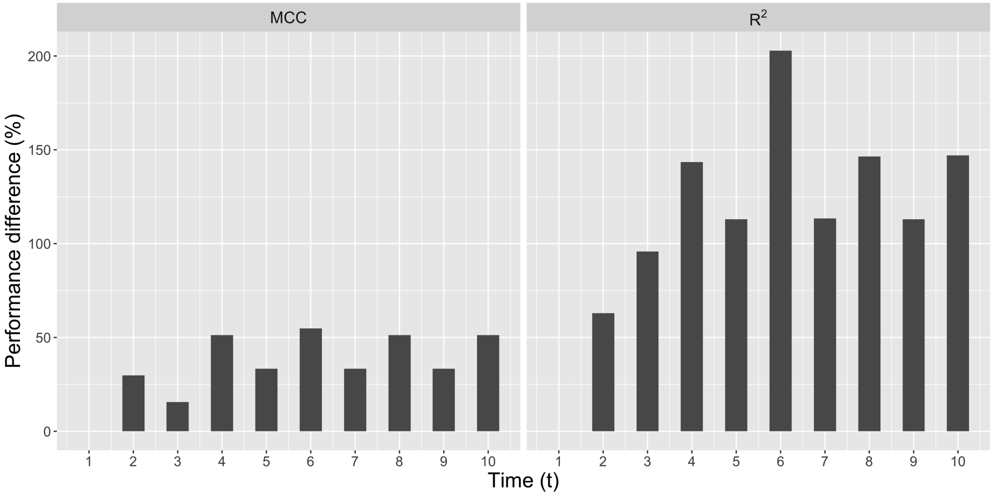


(c) $M = [1, 7]$.

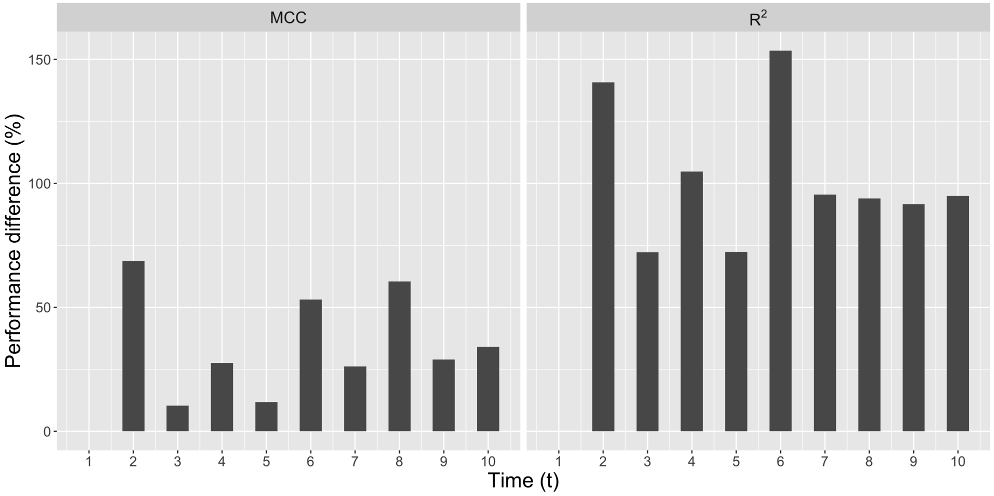
Figure 4.12: Performance differences between Models 2.a and 2.b for different values of M.



(a) $M = [1, 1]$.



(b) $M = [1, 4]$.



(c) $M = [1, 7]$.

Figure 4.13: Performance differences between Models 3.a and 3.b for different values of M.

4.4.2 Discussion

The integrated automata characteristics tend to be better predictors of long-term perturbation spreading behavior than the local automata characteristics, with larger M and t, because they tend to contain more information about the ECAs accumulated via the integration procedure. Since this is the case for ECAs—the simplest kinds of BNs—we can further hypothesize that the integrated characteristics of a broader classes of BNs where heterogeneity in other BN characteristics may also be predictive of the longer-term dynamical regimes. Moreover, the above results were derived for small ECAs with just $N = 10$ cells each which we were able to accurately characterize using the integrated measures. That is, we were able to compute the actual $F'^{(\tau)}$ at each τ for all the ECAs. However, as mentioned above, the computational costs quickly become prohibitive for larger networks. Thus, we can ask if the integrated characteristics of a BN would still be more predictive than the local characteristics if the integrated BNs were sampled. We will test these hypotheses in future work, although in the next section we demonstrate that even with sampling integration helps discern useful properties of a BN, in particular its "mechanism" across timescales.

Our original motivation for considering integrated BNs beside the original BN, as stated above, was that the former contained more information than the latter. We have demonstrated this for a fact for small ECAs, which may seem like a trivial result since it is clear in many ways that integration does produce more information. However, the experiments above show, somewhat counter-intuitively, that not all characterizations of the integrated BNs are more predictive than the same characterizations of the original BN. For example, we can see from Figs. 4.8(a) and 4.12(a) that k_e^7 and $p^7(1-p^7)$ are poorer classifiers (lower MCC) of dynamical regime at $t = 7$ than k_e^1 and $p^1(1-p^1)$ respectively for $M = [1, 1]$; only the combined $k_e^7 p^7(1-p^7)$ is better than $k_e^1 p^1(1-p^1)$. Thus, even though the integrated BNs are more informative, what characterizations we choose of the same as predictors of some dynamical property is not trivial. What we have shown in our experiments above show that there exist characterizations of the integrated BNs based on k_e and p that can be used as predictors of dynamical regimes at different timescales. A related arena for future work would revolve around the following question: what is an *optimal* timescale τ_{opt} such that a BN, $B^{\tau_{\text{opt}}}$, is sufficiently predictive of dynamics at all timescales $\tau > \tau_{\text{opt}}$? If it exists, an optimal timescale points to a form of timescale-based redundancy in dynamics. By exploring the τ_{opt} of different biochemical systems, we can thus explore how much timescale-based redundancy exist in them and what their implications are on evolution and control.

The overall motivations for integrating a BN are two-fold: (1) predict long-term dynamical regime by using BN characterizations as predictors; and more generally, (2) explain dynamics at any given timescale using a BN. We demonstrated point (1) with the help of the experiments on the ECA ensemble. However, for the small ECAs we have considered here, we can compute the full STG from which we can directly measure the dynamical regimes at any t rather than predict them using the integrated canalization properties of the ECAs. This takes us to point (2), which was touched upon with the help of the examples described in § 4.1 and 4.2.2. It refers to the broader objective of relating the description of a BN and the associated long-term dynamics. That is, we are not only interested in predicting the long-term behavior, or more generally behavior at an arbitrary timescale, both of which can be measured off of the STG, but to predict it using the properties of a BN operating at that timescale. Moreover, for larger BNs where computing the full STG is impossible, it may be possible to compute integrated BNs. However, as we noted earlier, the computational cost of integration also grows super-exponentially with BN size. If it is impossible to compute either, then we can use sampling techniques to estimate either, and the cost of sampling may depend on the technique used. Therefore, the question of whether it is less or more expensive to integrate a BN compared to computing the STG would have to be settled in future work with a proper comparison of the respective costs, with or without sampling.

The important point is that we need integrated BNs because they provide additional insights over the corresponding original BNs, in that they explicitly state how each automaton is regulated at a particular timescale. To illustrate it, suppose that we have a BN B, and we are interested in what explains the dynamics at timescale $\tau = 100$. We can answer that question by iterating B 100 times, which generates the behavior after passing through many intermediate configurations. On the other hand, by iterating the integrated BN B^{100} just once we can circumvent those intermediate configurations to generate the dynamical transitions that occur at that timescale. That is, we say that B^{100} *directly* describes the dynamics at $\tau = 100$. More importantly, the BN, B^{100}, may be very different from B itself[9], implying qualitatively different dynamics occurring over that timescale. For example, at timescale τ_1 two automata may be connected but not at a different timescale τ_2, implying that perturbations may propagate between them in τ_1 time steps but not in τ_2 time steps (just as in the example BN described in § 4.2.2). In this way, we can bridge the gap between the descriptions of the BN and the dynamics at any timescale.

The integrated BNs may further be useful in providing solutions to questions such as, "how to modify the logic rules of subsets of automata in a BN so that perturbations cannot drive a configuration to undesired attractors [23, 139]?". Specifically, the BN, B^{τ^*}, where τ^* is the asymptotic timescale, can be used to directly compute whether a perturbation to a set of automata will propagate to another set of automata of interest after τ^* time steps. By working backwards, this information could be used to logically deduce which logic rules of B may have to be modified and in what way, if the perturbations must be contained. Exactly how this can be achieved will be explored in future work.

Another upshot of motivation (2) above is that integration may also be used as an approach to explain the dynamic behavior across timescales of a BN. In the case of ECA Rule 30, we observed

---

[9] In the sense that B^{100} is typically not analytically predictable from B since it is a nonlinear dynamical system. For a linear system A, on the other hand, A^t for any t is analytically predictable since A would just be a matrix.

that its chaotic behavior may be explained by the corresponding rise in its k_e^t, the rate of which is almost the largest in the full ensemble of ECAs. This suggests that the integration procedure may help us discern the actual mechanism by which chaos[10] is generated in the behavior of Rule 30. In a similar vein, integration may provide complementary, if not deeper, explanations for why certain CAs perform better in say, the density classification tasks, than others [89], as we show in the next section.

4.5 Integration as an approach to elucidating mechanism

In this section, we demonstrate how integration can be used to elucidate the differences between the mechanisms of CAs. We consider CAs that attempt to solve the problem of the density classification task (DCT), where the goal of the CA is to deduce the majority (higher density) among the 1s and 0s in a given initial configuration of the CA [97]. If there are more 1s than 0s in an arbitrary initial configuration, then the CA must converge to a configuration with all 1s, else converge to all 0s.

Various CA rules designed to solve the DCT problem exist, and they are either constructed "by hand" or artificially evolved [95, 96]. The traditional approaches to decipher the dynamic mechanisms of the CAs focus on formulating minimal descriptions of how patterns propagate and how information is processed over space and time in the CA. These methods can be broadly classified into three categories: computational mechanics [29, 55, 95], information dynamics [43], and canalization based approaches [87]. In the computational mechanics framework, the dynamics of the CA is simplified in terms of 'regular domains' (repeating patterns of 1s and 0s), combinations of which known as 'particles', and a 'particle rule table' (PRT) containing a set of "laws" that describe how

---

[10] It has been observed [81] that Rule 30 maximizes a measure of 'information modification'; an interesting future project would be delve further into the relationship between this and our observation of the asymptotic maximization of k_e^t.

the particles of the CA interact. Indeed, it has been shown that the asymptotic behavior of a CA can be computed with high accuracy using just the corresponding PRT [55]. In the information dynamics framework, the mechanism is described in terms of spatiotemporal storage, transfer and modification of Shannon information. The canalization-based approach supplements the computational mechanics framework by relating the functional redundancy (or significance) of the regular domains in the PRT to the patterns of redundancy and symmetry in the CA's rule table (LUT).

Even though the individual mechanisms of the CAs have been elucidated using the frameworks above, a key challenge remains: how to attribute the differences in performance of the CAs to their individual descriptions (LUT or PRT) in a quantitative manner? To answer this question, it is crucial that we are able to compare the different mechanisms. PRTs pose a challenge here, since it may be difficult to compare them if, say, they contain different numbers of particles and qualitatively different interaction laws. Even if the PRTs are similar, as is the case [96] with rules 'ϕ_{par}' and 'GKL', it is not clear how that might explain performance difference. A comparison of LUTs, on the other hand, is possible via comparison of the corresponding schemata sets. For example, in [87], it was found that the rule named 'GP' contained a two-symbol schema that resulted in the switching of the cell state whereas another rule named 'GKL' contained none, thus indicating more input symmetry in the former. However, it is not clear how the local difference in the LUTs contributes to the overall performance difference.

To address this challenge, we propose an extension of the approach adopted in [87], where we compute the sequence of the integrated logic rules, $F'^{(\tau)}_{1(1)}, \forall \tau = 1, \ldots, \tau^*$; in the original approach, only $\tau = 1$ was considered. By focusing on patterns of regulation (logic rule), rather than activity, this approach facilitates comparisons of the strategies of different CAs via direct comparisons of the integrated logic rules (or some characterization of the same) at various timescales. As we show below, this approach also helps generate hypotheses or even explanations of the observed performance differences.

To demonstrate the utility of our approach, we investigate the performance difference between two different CA rules for the DCT problem, namely 'GKL' (a human-derived rule) and 'GP' (discovered through genetic programming) [87], by comparing their integrated logic rules. Figs. 4.14 displays the F_1', F_1'' (the corresponding sets for output 0 are symmetrical to these sets obtained by reflecting each schema about the center and negating the literals) and the canalization measures of the two rules. Fig. 4.15 depicts the schemata visually, showing that each cell in the CA has a neighborhood size of 7, receiving inputs from itself, three cells on the left and three on the right. We observe that the two rule sets are quite similar—a point elaborated in [87]. More importantly, both rules have the same k, p, k_e and k_s, and only their k_u differ, with GP containing slightly more input symmetry.

$$F_1' = \begin{Bmatrix} (1,\#,\#,1,\#,\#,\#), \\ (1,\#,\#,\#,1,\#,1), \\ (\#,\#,1,1,\#,\#,\#), \\ (\#,\#,1,\#,1,\#,1), \\ (\#,\#,\#,0,1,\#,1) \end{Bmatrix} \quad F_1'' = \begin{Bmatrix} (\#,\#,\#,0,1,\#,1), \\ (\#,\#,1,\dot{1},\dot{1},\#,\dot{\#}), \\ (1,\#,\#,\dot{1},\dot{1},\#,\dot{\#}), \\ (\dot{1},\#,\dot{\#},1,\#,\#,\#) \end{Bmatrix}$$

$k = 7$
$p = 0.5$
$k_e = 2.25$
$k_s = 1.875$
$k_u \approx 1.62$

(a) GKL

$$F_1' = \begin{Bmatrix} (1,\#,\#,\#,\#,\#,1), \\ (1,\#,\#,1,\#,\#,\#), \\ (\#,\#,\#,0,1,\#,1), \\ (\#,\#,1,\#,1,\#,1), \\ (\#,\#,1,0,\#,\#,1) \end{Bmatrix} \quad F_1'' = \begin{Bmatrix} (\dot{1},\#,\dot{\#},0,\dot{\#},\#,1), \\ (\dot{1},\#,\dot{1},\#,\dot{\#},\#,1), \\ (1,\#,\#,\dot{1},\#,\#,\dot{\#}) \end{Bmatrix}$$

$k = 7$
$p = 0.5$
$k_e = 2.25$
$k_s = 1.875$
$k_u \approx 1.59$

(b) GP

Figure 4.14: The input schemata, the two-symbol schemata sets, and the canalization measures of the GKL and GP rules. The constitution of F_1'' of the GKL rule is discussed in Appendix. C.

We consider two single-dimension CAs, one defined by the GKL rule and the other GP, containing $N = 21$ cells each. Fig. 4.16 depicts a sample behavior of the two CAs with the same initial configuration containing 11 ones—the minimum number required to constitute a majority. As can be seen, GKL is able to classify the majority by converging to a state with all 1s, whereas GP is unable to. To measure the performance of each CA, we computed the mean number of 1s in

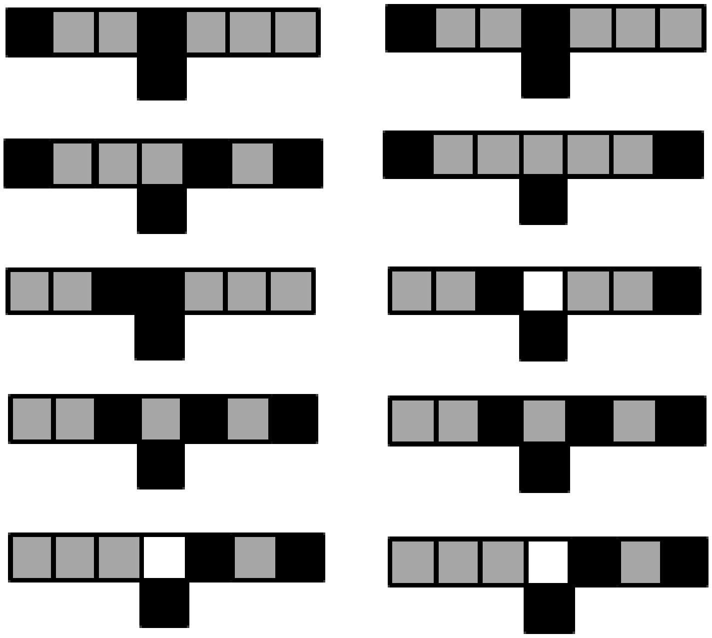
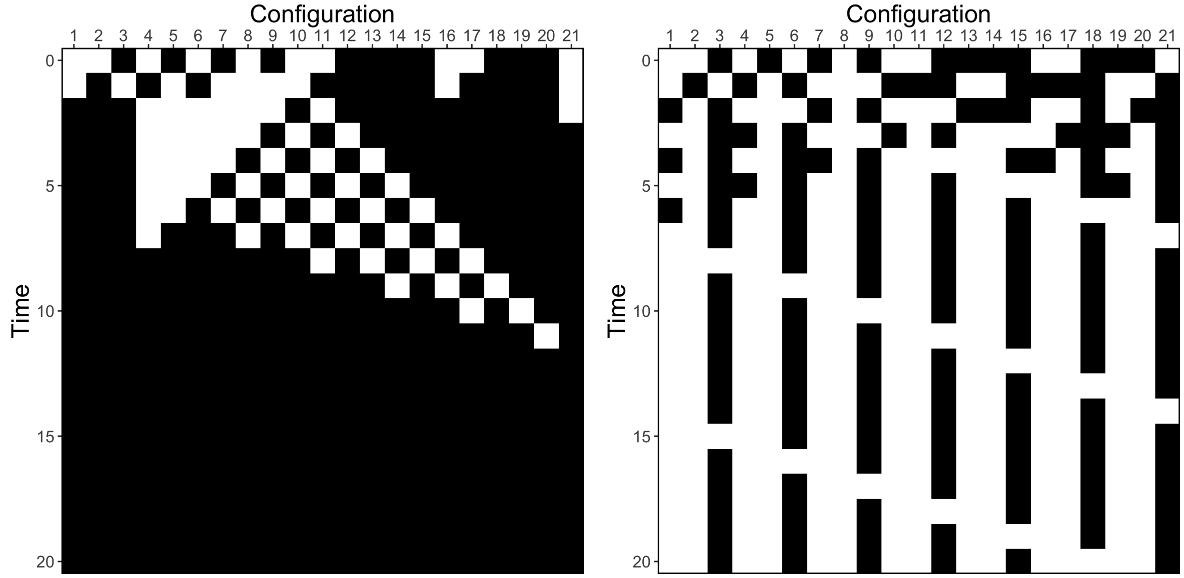

(a) GKL **(b)** GP

Figure 4.15: The input schemata sets of the GKL and GP rules in visual form.

a configuration $\langle n_1^t \rangle$ at each time step $t \in \{1, \ldots, 20\}$ (dynamics typically converge by $t = 20$) given an initial ($t = 0$) number of 1s $n_1^0 \in \{11, \ldots, 15\}$ (the performances are about the same for $n_1^0 > 15$). We considered all possible initial configurations for each n_1^0. We only considered the case of classification of majority 1s and not 0s since both rules treat 1s and 0s symmetrically, as mentioned above. Fig. 4.17 displays the $\langle n_1^t \rangle$ of the two CAs at each t, where we can see that GKL performs better than GP at every t. Although, note that GP has been previously known to perform better than GKL [87] in larger CAs containing 149, 599 and 999 cells. Such large CAs are thought to be necessary for DCT because they need sufficient time to process information and make a collective decision. Thus, the 21-cell CA may not really be capturing the essence of DCT, yet we can characterize their behavior in a DCT-like task, as described below.

What explains the performance difference between the two CAs? One hint comes from the differences in their dynamic behaviors, as captured by the respective PRTs: while GKL has just six



(a) GKL **(b)** GP

Figure 4.16: The behaviors of CAs with GKL and GP rules given an initial configuration with 11 ones which is the same for both CAs.

particles and its behavior can be summarized as successively classifying local densities with the locality range growing with time [95, 96], GP contains many more particles (estimated to be a minimum of 90 [87]). However, how the performance difference can be quantitatively attributed to the differences in PRTs is not clear. Another hint comes from the fact that the two rules are different at both the LUT and PRT levels [87], but again how these differences contribute to the overall performance difference is still an open question.

A step toward answering the above question was taken in [87], where the difference between the F' sets corresponding to the two rules was used to partly explain why the corresponding PRTs seem very different despite the high similarity between the LUTs. Here, we take that approach a step further, and compare the $F'^{(\tau)}$ sets of the two rules over a range of τ. The goal is to quantitatively state qualitative descriptions of CA dynamics such as "local densities getting successively classified", thereby explaining the performance differences. For the comparison of GKL and GP, we achieved it as follows.

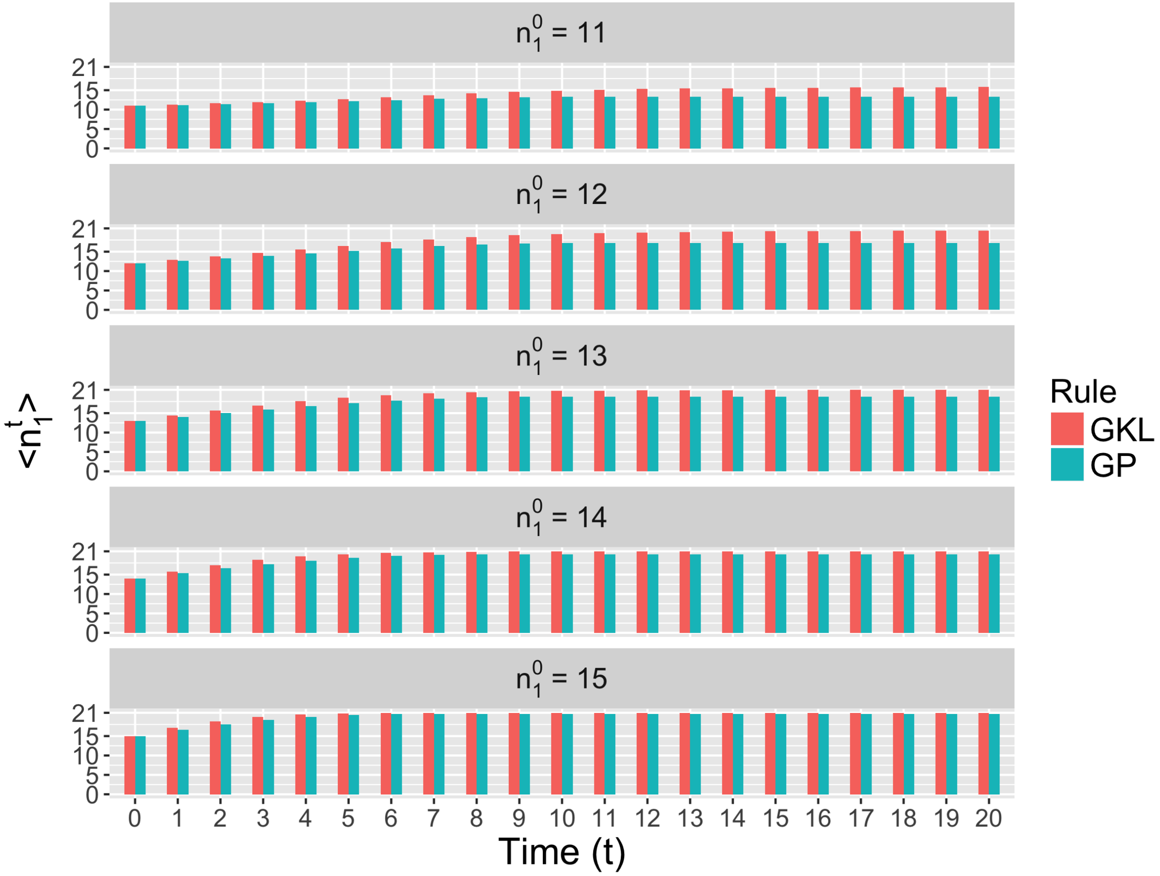


Figure 4.17: The performances of CAs with GKL and GP rules as measured by $\langle n_1^t \rangle$ at each $t \in \{1, \ldots, 20\}$ for each $n_1^0 \in \{11, \ldots, 15\}$. All possible $C(N, n_1^0)$ initial configurations for a given n_1^0 were considered.

For each CA, we computed the BFS $\left(F'^{(1)}_{1(1)}, F'^{(2)}_{1(1)}, \ldots F'^{(20)}_{1(1)}\right)$, which we refer to as the CA's *strategy*. We computed the BFS for a single cell only (cell number 1, to be precise), since all cells have the same LUT which implies that their LUTs at any given τ will also be the same. Furthermore, since we are only concerned with the case of classifying 1s and not 0s, we only computed the BFS associated with the state $x_1 = 1$ (the complementary BFS can be easily obtained through symmetry operations).

Due to the high computational cost of integration, we sampled the input schemata during the "compression" steps at every τ in the algorithm described in § 4.2.1. Specifically, we considered a maximum of 100 randomly chosen input schemata in the set being compressed. Through such random sampling, we obtained a set of 20 *estimated* strategies (BFS). We then collated the 20 BFs into a single BFS, where we simply concatenated the 20 different sets of $F'^{(\tau)}_{1(1)}$ at each τ and further compressed them—we refer to it as the *collated* strategy.

We considered the following characterizations of a given strategy (estimated or collated) of a CA at each timescale:

1. Estimated mean effective connectivity of a cell at timescale τ: k_e^τ.

   Since $F'^{(\tau)}_{1(1)}$ is an estimate obtained through random sampling, the k_e^τ computed on that set is also an estimate. Moreover, even though we only use $F'^{(\tau)}_{1(1)}$ ($F'^{(\tau)}_{1(0)}$ is also required) to compute the estimated k_e^τ, it won't further worsen the estimate, since $F'^{(\tau)}_{1(0)}$ is fully symmetrical to $F'^{(\tau)}_{1(1)}$, as explained above.

2. Estimated mean and standard deviation of the number of effective 1s in an initial configuration that turn ON a cell at time τ: $\mu^\tau_{n_e^1}$ and $\sigma^\tau_{n_e^1}$.

   These quantities respectively denote the mean and standard deviation of the set $T^\tau_{n_e^1}$ consisting of the minimum number of literal 1s in the redescribing input schema containing the maximum number of wildcards for each LUT entry $f^\tau_\alpha \in F^\tau_1$:

   $$T^\tau_{n_e^1} = \{n_e^1(f_1^\tau), n_e^1(f_2^\tau), \ldots\}$$

   where,

   $f^\tau_\alpha \in F^\tau_1$ denotes an entry in the LUT at timescale τ, and

   $$n_e^1(f^\tau_\alpha) = n_v^1, \text{where, } v = \underset{v:f^\tau_\alpha \in \Upsilon^\tau_v}{\arg\max}\,(n_v^\#), \text{where,}$$

   $n_v^\#$ is the number of wildcards in input schema v and $\Upsilon^\tau_v \in F'^{(\tau)}_1$.

   Notice that $\mu^\tau_{n_e^1}$ is calculated analogous to k_r (§ 1.6.3), the only difference being that the former considers the number of literal 1s in a schema whereas the latter considers the number of wildcards. Following is an example illustrating the above calculations. Let $F_1^\tau = \{(1,0,0),(1,0,1),(1,1,0),(1,1,1),(0,1,1),(0,0,0)\}$ denote the subset of LUT entries that go to output 1 at some timescale τ and let $F_1'^{(\tau)} = \{(1,\#,\#),(\#,1,1),(\#,0,0)\}$ be the associated input schemata set. Then, the corresponding $T^\tau_{n_e^1} = \{1,1,1,1,2,0\}$ where each $n_e^1(f_\alpha^\tau)$ is listed in the same order as f_α listed in F_1^τ above. Therefore, $\mu^\tau_{n_e^1} = 1$ and $\sigma^\tau_{n_e^1} \approx 0.63$.

Figs. 4.18 and 4.19 respectively display the above characterizations of the collated and estimated strategies of GKL and GP. For the estimated strategies, we calculated $\mu^\tau_{n_e^1}$ and $\sigma^\tau_{n_e^1}$ on a single $T^\tau_{n_e^1}$ obtained by concatenating the 20 $T^\tau_{n_e^1}$ sets. A comparison of the two figures suggests that the characterizations are stable across sample strategies, as conveyed by the small confidence intervals of k_e^τ and high similarity between the $\mu^\tau_{n_e^1}$ and $\sigma^\tau_{n_e^1}$ of the two figures.

We observe the following:

1. As the timescale increases ($\tau > 14$), the behaviors of the CAs become statistically distinct, where the corresponding $T^\tau_{n_e^1}$ separate (standard deviations barely overlap). More importantly, GKL displays a strategy with a lower $\mu^\tau_{n_e^1}$ and a larger $\sigma^\tau_{n_e^1}$ compared to that of GP. Furthermore, while the lower end of $\sigma^\tau_{n_e^1}$ of GKL covers the most extreme (bordering on ambiguity) case of $n_1^0 = 11$, GP touches $n_1^0 = 15$. This explains why GKL displays an asymptotically better behavior than GP for $n_1^0 \in \{11,\ldots,15\}$.

2. GKL is initially ($4 < \tau < 10$) more stable than GP (lower k_e^τ), but it asymptotically ($\tau > 16$) is more chaotic (higher k_e^τ). This roughly corresponds with the observation that $\sigma^\tau_{n_e^1}$ is initially smaller for GKL, but it asymptotically becomes larger. This makes sense because a greater sensitivity to a larger range of number of effective 1s would mean a larger k_e^τ.

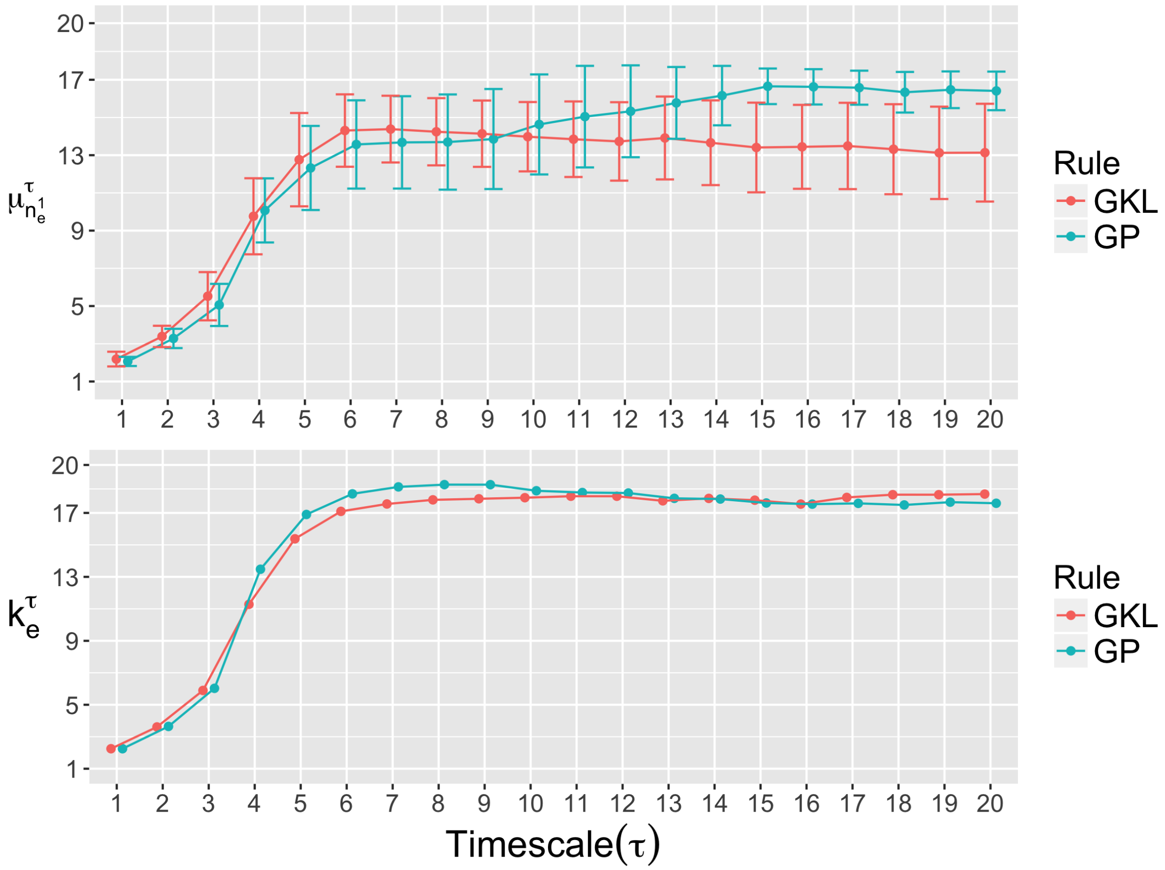


Figure 4.18: Characterizations of the collated strategies of GKL and GP. **(Top):** $\mu^{\tau}_{n_e^1}$, where the bars represent one standard deviation $\sigma^{\tau}_{n_e^1}$; **(Bottom):** k^{τ}_{e}. Both characterizations were computed over a range of $\tau \in \{1, \ldots, 20\}$.

4.5.1 Discussion

Together the above observations suggest different overall integration strategies for GKL and GP: while the former integrates smaller ranges of effective 1s into larger ranges, the latter integrates larger ranges of effective 1s into smaller ones asymptotically. The consequence is that GKL is asymptotically able to not only a larger range of effective 1s but also tune that range appropriately (covering the minimum number of n_1^0 for a majority). An ideal strategy for a perfect classifier would be for it to asymptotically evolve the 'majority function' [28], where the output would be 1 if and only if the number of 1s in the LUT entry is more than the number of 0s. However, it is known that

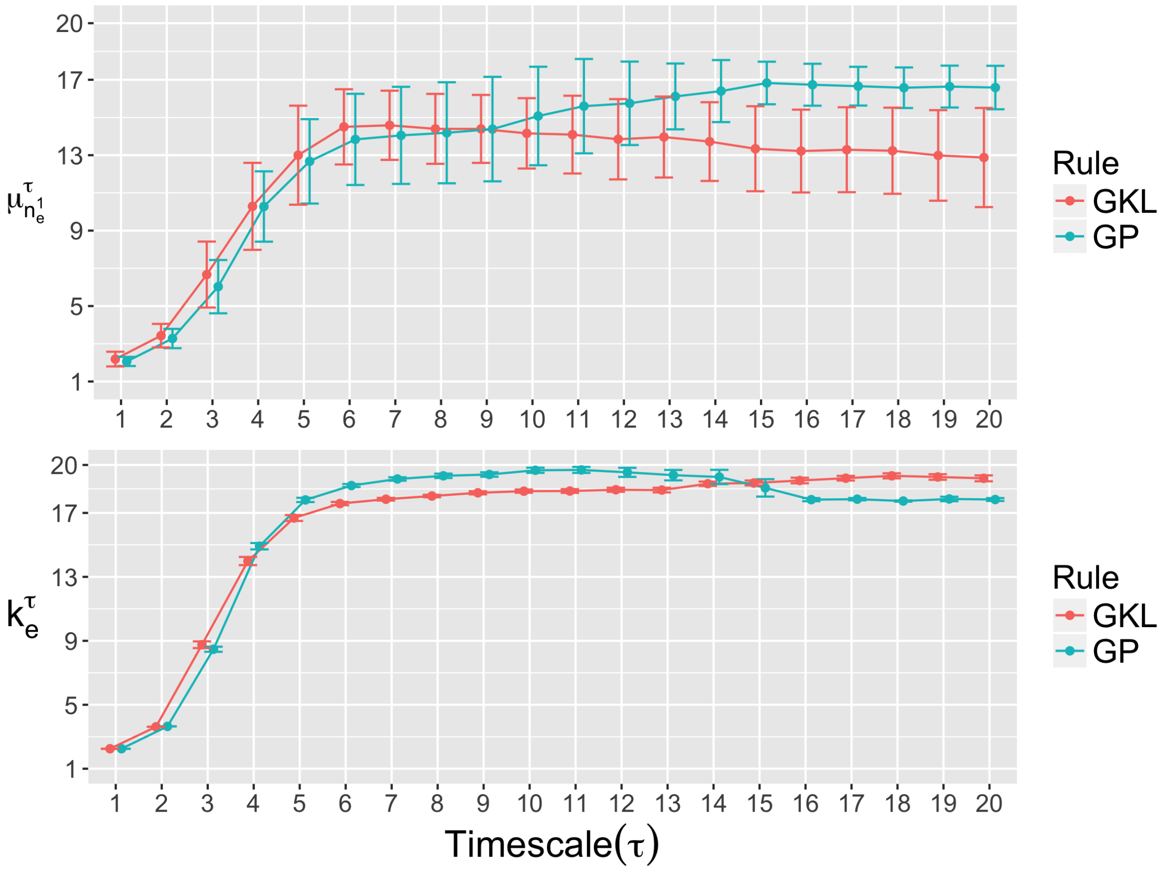


Figure 4.19: Characterizations of the estimated strategies of GKL and GP. **(Top):** $\mu^{\tau}_{n^1_e}$ computed over the concatenated set of $T^{\tau}_{n^1_e}$ corresponding to each of the 20 individual estimated strategies, where the bars represent one standard deviation $\sigma^{\tau}_{n^1_e}$; **(Bottom):** mean k^{τ}_e computed over the set of k^{τ}_e values computed on each of the 20 individual estimated $F'^{(\tau)}_{1(1)}$ at each τ; the bars represent the 95% confidence interval of the mean. Both characterizations were computed over a range of $\tau \in \{1, \ldots, 20\}$.

there does not exist a single-dimensional CA that perfectly solves the DCT problem [72]. Thus, another approach to analyzing the strategy of the above CAs is to measure the proximity of the LUTs with the majority function at each timescale, or use it as a fitness function measure to evolve rules (future work).

The above observations further suggest a hypothesis that a strategy of the form displayed by GKL may be necessary for any CA rule to perform relatively well. We can test it by considering other known CA rules [96] for the DCT problem and conduct a similar analysis as above (future work). Even though our analysis has helped discern the basic differences in the strategies of GKL and GP and explain why the former performs better, it is far from complete. In particular, we have only considered the strategy of a single cell in the CAs. We can also consider the *joint* strategies of sets of cells. Such an analysis would be required to explain why GKL performs better than GP also during the early time periods, since as far as a single cell is concerned the statistical behaviors of GKL and GP with respect to $\mu^{\tau}_{n^1_e}$ are similar.

Chapter 5

Conclusion

In this concluding chapter, we summarize the contributions made by this dissertation, discuss them in a broader context and describe avenues for future research.

5.1 Contributions

The following is a summary of the key contributions of this dissertation:

1. **Effective connectivity is a better predictor of criticality of homogeneous RBNs than in-degree.**

   With the help of numerical simulations, I showed that a theory of criticality based on k_e given by $3.93.\langle k_e \rangle .p(1-p) = 1$ for homogeneous RBNs with $k \leq 8$ is better the current best theory of criticality for homogeneous RBNs given by $2kp(1-p) = 1$ [67]. I also showed that $\langle k_e \rangle$ is a better characterization than $\langle k \rangle$ of the dynamical regimes of Systems biology models even though they are not as homogeneous as the RBNs we considered in the simulations.

   In both cases, the main reason why a criticality model based on k_e is better than one based on k because the former is more nuanced and thus informative than the latter. We illustrated this with the help of an approximation, where we showed that for the same k and p, increasing the

k_e of an automaton increases its expected average input activity which is related to criticality.

2. **Unified canalization is a better predictor of criticality of homogeneous RBNs with large in-degrees than effective connectivity.**

   I formulated a new measure of unified canalization (k_u) by measuring both effective connectivity and input symmetry with a common yardstick of "hypercube dimension", thus combining them into one. With the help of numerical simulations, I showed that a slightly better prediction of criticality than the one based on k_e above is given by $4.85.\langle k_u \rangle .p(1-p) = 1$ for homogeneous RBNs with $k \leq 12$, where k_u is a measure of canalization that unifies effective connectivity and input symmetry into one. I further showed that the efficacy of $\langle k_u \rangle$ as a predictor of criticality for subsets of RBNs conditioned on k increases with k. Finally, I showed that $\langle k_e \rangle$ is a better characterization than $\langle k \rangle$ of the dynamical regimes of Systems biology models even though they are not as homogeneous as the RBNs we considered in the simulations.

   The main reason why k_u provides a better performance at large k is that more room is created for input symmetry at large k. I illustrated this with the help of a simple calculation that shows that the proportion of two-symbol schema signatures quickly rises as the dimension (k) of the hypercube increases. I also explained why a criticality model based on $\langle k_u \rangle$ performs worse than one based on $\langle k_e \rangle$ for RBNs with $k \leq 8$: at low k, effective connectivity and input symmetry are better off being separate predictors rather than unified, as their relative contributions are inherently different (thus forcing them to be the same in the single-parameter model reduced the performance).

3. **Integration yields more accurate predictors of criticality at an arbitrary timescale of the dynamics, and elucidates the mechanism across timescales.**

   I designed an algorithm for integrating a BN that iteratively composes the set of logic rules of the BN with itself. With the help of numerical simulations, I showed that the mean k_e of an integrated elementary cellular automaton (ECA) that generates the dynamics observed at timescale τ is a better predictor of criticality of the ECA at time step τ than the mean k_e of the original ECA. I also showed that integration can elucidate the mechanism of a cellular automaton (CA), by comparing various characterizations (including k_e) of the integrated CAs of two types of CA rules namely 'GKL' and 'GP' used for the density classification task (DCT). Specifically, I showed that while GKL integrates smaller ranges of effective 1s (in the initial configuration of the CA) at smaller timescales into larger ranges at larger timescales, GP does it slightly the other way around.

The main reason why integration both improves prediction of criticality at an arbitrary timescale and elucidates the evolution of mechanism across timescales is that it identifies the regulatory logic that underlies the dynamics observed at a given timescale.

5.2 Discussion and Outlook

One of the central challenges in the theory of complex systems is understanding the relationship between the description of a complex system (e.g., structure, logic etc.) and its dynamics (e.g., phase space structure, attractor statistics, dynamical regime etc.). In this dissertation, we focus on the relationship between the static characteristics of a BN (a generic nonlinear discrete dynamical system) and its dynamical behavior. A clear understanding of the relationship between the two would ultimately enable us to better control and influence the dynamics via appropriate "tinkering" of the BN. The current theory for predicting the dynamical regime of a homogeneous RBN is given by $2kp(1-p) = 1$ [67]: if the value of $2kp(1-p) > 1$ the BN is expected to be chaotic, and stable otherwise. Thus, one way to make a chaotic BN stable is to reduce its k by removing edges and (or) $p(1-p)$ by altering the logic rules. In order to improve the prediction of dynamical regime, based on some characterization of the BN, it is necessary that the characterization is more informative and nuanced than current measures. In this dissertation, that new information is provided by two sources: (1) canalization and (2) integration. Canalization provides the basis for computing a more accurate characterization of the connectivity of a BN than those based on the apparent network structure (Chapters 2 and 3). Integration provides the basis for computing a more accurate BN that operates, or determines the dynamics, at a given timescale of interest (Chapter 4).

In Chapter 2, I showed that k_e, a measure of partial canalization, is not only more informative than k but also more predictive of criticality than k. For this reason, we expect that k_e would remain a better predictor of criticality than k even for RBNs with $k > 8$, the maximum k considered in Chapter 2. Moreover, it would be interesting to see how the results we have for homogeneous RBNs extend to heterogeneous RBNs. The important question there would be how the heterogeneity in the distribution of k reflects in the corresponding distribution of k_e. It may be that a scale-free distribution of k may translate into a less heterogeneous, not-so-scale-free, distribution of k_e, in which case one can question if the network is indeed scale-free. Another question there would be how the mean k_e changes if the mean k is preserved while transforming the distribution of k into a scale-free one. This is an important question because it was found in [45] that increasing the heterogeneity of k while preserving its mean made the BN more stable. One possible explanation for this phenomenon is that the mean k_e was reduced, making the BN more stable.

Comparison of k_e with senstivity (s) and the more recently introduced 'c-sensitivity' [61] as predictors of criticality shall constitute another important line of investigation for future work. We briefly discussed in Chapter 1 why s may be considered as a linear measure, and k_e nonlinear in comparison. It can be shown that $\langle s \rangle = \widehat{D}_1^1/N$. That is, the average sensitivity of the BN is equal to the expected perturbation size after one time step if exactly one automaton is initially perturbed; such a perturbation can be visualized as a tree, meaning an automaton receives an input perturbation from at most one other automaton, which is why the linearity of $\langle s \rangle$ enables an exact calculation of $\widehat{D}_1^1/N$. For larger initial perturbation sizes and (or) longer periods of time, an exact calculation with $\langle s \rangle$ typically becomes impossible since multiple automata come into play, and the perturbations may not be propagating on a tree anymore. That is, an automaton may receive input perturbations from multiple other automata at any given time step, and the probability that the automaton is perturbed under such conditions is not captured by s. c-sensitivity, on the other hand, indicates the probability that an automaton flips its state if c of its inputs flip. Thus, the full k-vector of c-sensitivities of an automaton with k inputs can be used to exactly calculate $\widehat{D}_M^1/N$ for any initial range of perturbation sizes M, hence D_c can therefore be exactly calculated as well [61]. The non-linearity of k_e lies in the fact that it summarizes the number of inputs that *jointly* canalize the output of the automaton, which can clearly be linked to the c-sensitivities since they capture the number of inputs that jointly perturb the output of the automaton. However, since the former is a scalar summary, it would be expected to contain less information than the latter. Thus, the main question in the comparison of k_e with the k-vector of c-sensitivities would be how much does the additional information contained in the latter buys in comparison with the former.

In Chapter 3, I showed that k_u, another measure of partial canalization, is not only more informative than k_e but also more predictive of criticality than k_e for RBNs with a large k (> 8). Since k_e is folded into k_u, the above discussion for k_e extends to k_u as well. An important question for future research in this realm is an exact explanation for why input symmetry becomes a more dominant predictor of criticality with larger k. I have hinted at a possible explanation in the chapter based on the fact that the proportion of two-symbol schema signatures rises exponentially with k, suggesting that k_u becomes more informative than k_e at large k. However, how that extra information actually translates into a better prediction of criticality requires a deeper examination. One way to address this question is to analytically estimate the c-sensitivities of an automaton from the corresponding set of wildcard schemata and the set of two-symbol schemata, and compare the two estimates with the actual c-sensitivities (since they can be used to exactly calculate D_c). By naturally considering the differences between wildcard schemata and two-symbol schemata with regards to how they respond to input perturbations, this calculation would reveal the exact manner by which the two types of schemata contribute differently to criticality.

A formal treatment of wildcard schemata and two-symbol schemata in future work would naturally involve the mathematical notion of symmetry groups and their properties. The study of symmetry in digital circuits dates back to Claude Shannon's period [115]. The symmetry properties of Boolean automata in particular have been well-studied: the finite group consisting of complementation and permutation operations on the k input variables of an automaton is isomorphic to the 'hyperoctahedral group', O_k, a symmetry group on the vertices of the corresponding k-dimensional hypercube [120]. Efficient algorithms to detect the *global* symmetries of an automaton, the set of symmetries that preserve the entire hypercube, based on the properties of O_k exist [120]. Wildcard schemata and two-symbol schemata, on the other hand, capture the *local* symmetries of an automaton, the set of symmetries that preserve only a portion of the hypercube. Treating them as formal symmetry objects may enable the formulation of more efficient ways of detecting the same. More importantly, it would reveal the fundamental differences between the two types of symmetries they respectively capture, thereby further elucidating the different ways by which they contribute to criticality.

Symmetry has also been observed to manifest in the dynamical behavior of the automata in critical RBNs [56]: automata with the same canalization properties (as they define it) have the same probability of being dynamically "active" (whose state is not frozen). The symmetry group that relates those automata is known as the "Zyklenzeiger" group that preserves canalization [56]. Since the critical RBNs were constructed as per the current theory of criticality in that work, the natural question would be how our new theories of criticality based on k_e and k_u might alter their results. It would also be interesting to investigate how the local symmetry groups (schemata) of the individual automata constrain the emergence of the symmetry group that relates sets of automata (such as those found in [56]) in critical RBNs. Moreover, phase transition phenomena are characterized by symmetry-breaking, a well-known example of which is the loss of rotational symmetry in the magnetic orientation of a ferromagnetic system as the driving temperature is lowered below a critical point, the Curie temperature. One way in which phase transitions manifest in RBNs is via the breaking of the Zyklenzeiger group [56, 79]. Since we have shown in this dissertation that symmetry controls phase transition, an important question would be how that is related to symmetry-breaking, a characteristic of phase transition. That is to ask if a unified theory of criticality, where symmetry is both the tuning parameter and the order parameter, exists.

Both Chapters 2 and 3 suggest that a more precise way of altering the dynamical regime of a BN is to modify the k_e or k_u of the logic rules of the automata rather than modifying k or p (something along the lines of [23, 139]). Moreover, for actual Systems biology models, which may not accurately fit a characterization of an RBN ensemble, it would be necessary to formulate precise alteration strategies depending on the observed distributions of k_e and k_u. Such a strategy may also consist of the priority or the order in which specific automata must be altered. The implications of our ensemble-based results on natural selection takes us back to Stuart Kauffman's original research program whose basic premise was that evolution worked at the level of the k of gene regulatory systems since it determined much of the system's characteristic macroscopic dynamics. In Kauffman's words, "order comes for free" via self-organization [67]. Our results suggest a refinement to Kauffman's hypothesis: natural selection may be working at the level of k_e or k_u rather than k. Although our theoretical results support this hypothesis, there are challenges that need to be addressed particularly in terms of the mechanism by which evolution may accomplish this. For instance, it would be natural to ask how canalization may be available as genetic parameters susceptible to mutations. Moreover, it would be clearly more expensive to tweak the canalization parameters than k, simply because the former takes more information and therefore more energy. That is, there would be a tradeoff between accuracy and energy expense for evolution to deal with, since tweaking canalization would yield more accuracy whereas tweaking k would take less energy.

In Chapter 4, I showed that an integrated BN is more informative than the original BN insofar as the dynamics at a given timescale is concerned. Thereby, I showed that the mean k_e of the former is more predictive of the dynamical regime at a given time than the mean k_e of the latter. The main reason why integration enables more accurate predictions is that nonlinear systems typically cannot be analytically solved. That is, the description of a nonlinear system (e.g., ODE, BN etc.) generally does not contain sufficient information about the future states of the system, which integration supplies. Even though integration is not necessary for predicting future behavior—analysis of the STG provides an alternative approach—it is the only approach that provides a BN-based prediction. In other words, while it is possible to predict, or even compute, dynamics at an arbitrary timescale based off of the STG, it is integration that provides the basis for making those calculations using the properties (e.g., canalization) of a BN. The reason is that integration identifies the regulatory logic that underlies the dynamics observed at that timescale. Thus, integration is central to bridging the gap between the descriptions of the BN and the dynamics at arbitrary timescales, which is one of the broader themes of this dissertation.

Although the concept of integration is not novel (the sequence of higher iterates of an iterated map is known as 'Picard sequence' in the literature [133]), this is the first time it is applied to the case of BN and the resulting integrated BNs characterized by k_e^τ. As we show in the example BN and the ensemble of ECAs, k_e^τ can be a nonlinear function of τ. This means that the number of effective inputs of an automaton at a longer timescale could be lower than its original k_e. One reason for this is the finiteness of the BN, which goes against the typical assumption of analytical approximations (e.g., 'annealed' approximation [67]) of the prediction of dynamical behavior that $N \to \infty$ whose net effect is that the number of inputs (effective or otherwise) an automaton could only increase with timescale. However, even in finite BNs with a regular structure, like CAs, number of inputs of a node could monotonously increasing with timescale. In the case of ECAs, it is easy to show that $k^{\tau+1} = k^{\tau} + 2$, that is, a cell receives inputs from two more cells with every consecutive timescale. This would suggest that k_e^{τ} is also a monotonically increasing function of τ in an ECA, which, as we have seen, is not necessarily the case. This is chiefly because of the non-linearity of the Boolean functions, which is the main reason why integration is necessary to identify how the actual logic and the corresponding k_e of a cell evolves with timescale.

By providing insights into the mechanism of a BN, integration may reveal the precise means by which "order comes for free" via self-organization [67]. Moreover, integration may provide important clues to the question of how to alter the logic of a BN so that it produces some desired behavior [23, 139]. That is, given a BN B and the goal of altering a particular behavior at some timescale τ, we say that the logic of B^{τ} may hold clues as to the required alterations to the logic of B. Even though in general there could be many possible B that can give rise to B^{τ} (hinting at irreversibility and degeneracy), given a B and a B^{τ} there is a unique path that connects them in either direction. Therefore, all deductions about alterations to B can be made starting with B^{τ} and working backwards. For example, suppose that the $k_e(x_i) < k_e(x_j)$ in B, and $k_e^{\tau}(x_i) > k_e^{\tau}(x_j)$ in B^{τ}, and that B^{τ} is observed to be chaotic, then we can deduce that x_i contributes more the chaotic behavior at τ than x_j. Hence, it may be that lowering the k_e of x_i rather than x_j, or some other automaton for that matter, is necessary to contain the chaos at τ. This is where the sequence of $(B^1, \ldots, B^{\tau})$ could help reveal the exact causes of the higher k_e of x_i at τ and thus provide precise prescriptions for how B could be altered in order to contain the chaos at τ. Exactly how this can be accomplished will be the subject matter of future work. Another possibility here is that we don't even need to compute B^{τ}, and that computing B^t for some $t < \tau$ is sufficient to deduce the causes of the chaotic behavior of τ; here t may be thought of as the "effective timescale" of B, and that every timescale greater than t is a "redundant timescale".

A common challenge to almost all of the pursuits of this dissertation is computational cost. While the high computational cost of computing wildcard schemata is well-known [28], computing two-symbol schemata, or computing the integrated wildcard schemata only adds to that cost. While efficient techniques have been explored in detecting global symmetry in Boolean hypercubes (e.g., [120]), research is needed to efficiently detect local symmetry as captured by the two-symbol schemata. Likewise, how integration can be done efficiently also needs deeper investigation, especially given the dearth of research in this area. Algebraic approaches for efficiently computing attractors and controllers exist (without an analogous step for integration), where a BN is represented as a system of polynomial equations [25, 74, 100]. One question we can ask here is if such a representation, or an approximation containing terms with a maximum degree of the polynomial, may reduce the computational cost of integration.

Bibliography

[1] Réka Albert. Boolean modeling of genetic regulatory networks. In *Complex networks*, pages 459–481. Springer, 2004.

[2] Réka Albert and Hans G Othmer. The topology of the regulatory interactions predicts the expression pattern of the segment polarity genes in drosophila melanogaster. *Journal of theoretical biology*, 223(1):1–18, 2003.

[3] Reka Albert and Juilee Thakar. Boolean modeling: a logic-based dynamic approach for understanding signaling and regulatory networks and for making useful predictions. *Wiley Interdisciplinary Reviews: Systems Biology and Medicine*, 6(5):353–369, 2014.

[4] Maximino Aldana. Boolean dynamics of networks with scale-free topology. *Physica D: Nonlinear Phenomena*, 185(1):45–66, 2003.

[5] Maximino Aldana, Enrique Balleza, Stuart Kauffman, and Osbaldo Resendiz. Robustness and evolvability in genetic regulatory networks. *Journal of theoretical biology*, 245(3):433–448, 2007.

[6] Maria I Arnone and Eric H Davidson. The hardwiring of development: organization and function of genomic regulatory systems. *Development*, 124(10):1851–1864, 1997.

[7] John Avery. *Information theory and evolution*. World Scientific, 2003.

[8] Pierre Baldi, Søren Brunak, Yves Chauvin, Claus AF Andersen, and Henrik Nielsen. Assessing the accuracy of prediction algorithms for classification: an overview. *Bioinformatics*, 16(5):412–424, 2000.

[9] Elizabeth Ball. *Dynamic Spread of Social Behavior in Boolean Networks*. PhD thesis, University of Nebraska at Omaha, 2011.

[10] Enrique Balleza, Elena R Alvarez-Buylla, Alvaro Chaos, Stuart Kauffman, Ilya Shmulevich, and Maximino Aldana. Critical dynamics in genetic regulatory networks: examples from four kingdoms. *PLoS One*, 3(6):e2456, 2008.

[11] Albert-László Barabási. Scale-free networks: a decade and beyond. *science*, 325(5939):412–413, 2009.

[12] Albert-László Barabási. The network takeover. *Nature Physics*, 8(1):14, 2011.

[13] Albert-Laszlo Barabasi and Zoltan N Oltvai. Network biology: understanding the cell's functional organization. *Nature reviews genetics*, 5(2):101–113, 2004.

[14] Marc Barthélemy, Alain Barrat, Romualdo Pastor-Satorras, and Alessandro Vespignani. Dynamical patterns of epidemic outbreaks in complex heterogeneous networks. *Journal of theoretical biology*, 235(2):275–288, 2005.

[15] Baruch Barzel and Albert-László Barabási. Universality in network dynamics. *Nature physics*, 9(10):673–681, 2013.

[16] John M Beggs. The criticality hypothesis: how local cortical networks might optimize information processing. *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 366(1864):329–343, 2008.

[17] Stefan Bornholdt. Boolean network models of cellular regulation: prospects and limitations. *Journal of the Royal Society Interface*, 5(Suppl 1):S85–S94, 2008.

[18] Stefan Bornholdt and Thimo Rohlf. Topological evolution of dynamical networks: Global criticality from local dynamics. *Physical Review Letters*, 84(26):6114, 2000.

[19] Katherine Brading and Elena Castellani. Symmetry and symmetry breaking. In Edward N. Zalta, editor, *The Stanford Encyclopedia of Philosophy*. Spring 2013 edition, 2013.

[20] Hector Corrada Bravo, Vasyl Pihur, Matthew McCall, Rafael A Irizarry, and Jeffrey T Leek. Gene expression anti-profiles as a basis for accurate universal cancer signatures. *BMC bioinformatics*, 13(1):272, 2012.

[21] J. Bruin. Faq: What are pseudo r-squareds? `http://www.ats.ucla.edu/stat/mult_pkg/faq/general/Psuedo_RSquareds.htm`, OCT 2011.

[22] Nicolas E Buchler, Ulrich Gerland, and Terence Hwa. On schemes of combinatorial transcription logic. *Proceedings of the National Academy of Sciences*, 100(9):5136–5141, 2003.

[23] Colin Campbell and Réka Albert. Stabilization of perturbed boolean network attractors through compensatory interactions. *BMC systems biology*, 8(1):1, 2014.

[24] Kátia K Cassiano and Valmir C Barbosa. Information integration in elementary cellular automata. *arXiv preprint arXiv:1410.7584*, 2014.

[25] Daizhan Cheng, Hongsheng Qi, and Zhiqiang Li. *Analysis and control of Boolean networks: a semi-tensor product approach*. Springer Science & Business Media, 2010.

[26] Tobias S Christensen, Ana Paula Oliveira, and Jens Nielsen. Reconstruction and logical modeling of glucose repression signaling pathways in saccharomyces cerevisiae. *BMC systems biology*, 3(1):7, 2009.

[27] Kevin A Clarke. A simple distribution-free test for nonnested model selection. *Political Analysis*, 15(3):347–363, 2007.

[28] Yves Crama and Peter L Hammer. *Boolean functions: Theory, algorithms, and applications*. Cambridge University Press, 2011.

[29] James P Crutchfield and Melanie Mitchell. The evolution of emergent computation. *Proceedings of the National Academy of Sciences*, 92(23):10742–10746, 1995.

[30] Dragoš M Cvetković, Peter Rowlinson, and Slobodan Simić. *An introduction to the theory of graph spectra*, volume 75. Cambridge University Press Cambridge, 2010.

[31] Jacques Demongeot, Hedi Ben Amor, Adrien Elena, Pierre Gillois, Mathilde Noual, and Sylvain Sené. Robustness in regulatory interaction networks. a generic approach with applications at different levels: physiologic, metabolic and genetic. *International Journal of Molecular Sciences*, 10(10):4437–4473, 2009.

[32] Bernard Derrida and Yves Pomeau. Random networks of automata: a simple annealed approximation. *EPL (Europhysics Letters)*, 1(2):45, 1986.

[33] Bernard Derrida and Dietrich Stauffer. Phase transitions in two-dimensional kauffman cellular automata. *EPL (Europhysics Letters)*, 2(10):739, 1986.

[34] Rick Durrett. Some features of the spread of epidemics and information on a random graph. *Proceedings of the National Academy of Sciences*, 107(10):4491–4498, 2010.

[35] Jacob Feldman. A catalog of boolean concepts. *Journal of Mathematical Psychology*, 47(1):75–89, 2003.

[36] Thomas Flatt. The evolutionary genetics of canalization. *The quarterly review of biology*, 80(3):287–316, 2005.

[37] Jeffrey J Fox and Colin C Hill. From topology to dynamics in biochemical networks. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 11(4):809–815, 2001.

[38] Jianxi Gao, Baruch Barzel, and Albert-Lászl̃ó Barabási. Universal resilience patterns in complex networks. *Nature*, 530(7590):307–312, 2016.

[39] Alexander J Gates and Luis M Rocha. Control of complex networks requires both structure and dynamics. *Scientific reports*, 6, 2016.

[40] C. Gershenson. Introduction to random boolean networks. In M. Bedau, P. Husbands, T. Hutton, S. Kumar, , and H. Suzuki, editors, *Workshop and Tutorial Proceedings, Ninth International Conference on the Simulation and Synthesis of Living Systems (ALife IX)*, pages 160–173, 2004.

[41] Clare E Giacomantonio and Geoffrey J Goodhill. A boolean model of the gene regulatory network underlying mammalian cortical area development. *PLoS Comput Biol*, 6(9):e1000936, 2010.

[42] Greg Gibson and Günter Wagner. Canalization in evolutionary genetics: a stabilizing theory? *Bioessays*, 22(4):372–380, 2000.

[43] Alan Godoy, Pedro Tabacof, and Fernando J Von Zuben. The role of the interaction network in the emergence of diversity of behavior. *PloS one*, 12(2):e0172073, 2017.

[44] Jelle Goeman, Rosa Meijer, Nimisha Chaturvedi, and Maintainer Jelle Goeman. Package 'penalized'. 2017.

[45] Alireza Goudarzi. On the effect of heterogeneity on the dynamics and performance of dynamical networks. 2012.

[46] Nancy R Gough and John F Foley. Focus issue: Unraveling signaling complexity. *Sci. Signal.*, 2(81):eg10–eg10, 2009.

[47] David J Hand. Measuring classifier performance: a coherent alternative to the area under the roc curve. *Machine learning*, 77(1):103–123, 2009.

[48] ES Harris, Bruce K Sawhill, Andrew Wuensche, Stuart Kauffman, et al. Biased eukaryotic gene regulation rules suggest genome behaviour is near edge of chaos. In *Santa Fe Institute*, pages 97–05, 1997.

[49] Stephen E Harris, Bruce K Sawhill, Andrew Wuensche, and Stuart Kauffman. A model of transcriptional regulatory networks based on biases in the observed regulation rules. *Complexity*, 7(4):23–40, 2002.

[50] Michael A Harrison. *Introduction to switching and automata theory.* Mac Graw-Hill, 1965.

[51] Inman Harvey. The microbial genetic algorithm. In *European Conference on Artificial Life*, pages 126–133. Springer, 2009.

[52] Tomas Helikar. Interactive modeling of biological networks. `https://cellcollective.org`, 2016.

[53] Tomáš Helikar, Bryan Kowal, Sean McClenathan, Mitchell Bruckner, Thaine Rowley, Alex Madrahimov, Ben Wicks, Manish Shrestha, Kahani Limbu, and Jim A Rogers. The cell collective: toward an open and collaborative approach to systems biology. *BMC systems biology*, 6(1):96, 2012.

[54] Franziska Herrmann, Alexander Groß, Dao Zhou, Hans A Kestler, and Michael Kühl. A boolean model of the cardiac gene regulatory network determining first and second heart field identity. *PLoS One*, 7(10):e46798, 2012.

[55] Wim Hordijk, James Crutchfield, and Melanie Mitchell. Mechanisms of emergent computation in cellular automata. In *Parallel Problem Solving from Nature—PPSN V*, pages 613–622. Springer, 1998.

[56] Shabnam Hossein, Matthew D Reichl, and Kevin E Bassler. Symmetry in critical random boolean network dynamics. *Physical Review E*, 89(4):042808, 2014.

[57] Sui Huang, Ingemar Ernberg, and Stuart Kauffman. Cancer attractors: a systems view of tumors from a gene network dynamics and developmental perspective. In *Seminars in cell & developmental biology*, volume 20, pages 869–876. Elsevier, 2009.

[58] DJ Irons. Logical analysis of the budding yeast cell cycle. *Journal of theoretical biology*, 257(4):543–559, 2009.

[59] Yasser Iturria-Medina, Roberto C Sotero, Paule J Toussaint, Alan C Evans, Alzheimer's Disease Neuroimaging Initiative, et al. Epidemic spreading model to characterize misfolded proteins propagation in aging and associated neurodegenerative disorders. *PLoS Comput Biol*, 10(11):e1003956, 2014.

[60] Kayse Jansen and Mihaela Teodora Matache. Phase transition of boolean networks with partially nested canalizing functions. *The European Physical Journal B*, 86(7):1–11, 2013.

[61] Claus Kadelka, Jack Kuipers, and Reinhard Laubenbacher. The influence of canalization on the robustness of boolean networks. *arXiv preprint arXiv:1607.04474*, 2016.

[62] Fredrik Karlsson and Michael Hörnquist. Order or chaos in boolean gene networks depends on the mean fraction of canalizing functions. *Physica A: Statistical Mechanics and its Applications*, 384(2):747–757, 2007.

[63] Stuart Kauffman, Carsten Peterson, Björn Samuelsson, and Carl Troein. Random boolean network models and the yeast transcriptional network. *Proceedings of the National Academy of Sciences*, 100(25):14796–14799, 2003.

[64] Stuart Kauffman, Carsten Peterson, Björn Samuelsson, and Carl Troein. Genetic networks with canalyzing boolean rules are always stable. *Proceedings of the National Academy of Sciences of the United States of America*, 101(49):17102–17107, 2004.

[65] Stuart A Kauffman. Metabolic stability and epigenesis in randomly constructed genetic nets. *Journal of theoretical biology*, 22(3):437–467, 1969.

[66] Stuart A Kauffman. Emergent properties in random complex automata. *Physica D: Nonlinear Phenomena*, 10(1):145–156, 1984.

[67] Stuart A Kauffman. *The origins of order: Self organization and selection in evolution.* Oxford University Press, USA, 1993.

[68] Stuart A Kauffman. *Investigations.* Oxford University Press, 2000.

[69] Gwenael Kervizic and Laurent Corcos. Dynamical modeling of the cholesterol regulatory pathway with boolean networks. *BMC systems biology*, 2(1):99, 2008.

[70] Artemy Kolchinsky, Alexander J Gates, and Luis M Rocha. Modularity and the spread of perturbations in complex dynamical systems. *Physical Review E*, 92(6):060801, 2015.

[71] Dmitry Krotov, Julien O Dubuis, Thomas Gregor, and William Bialek. Morphogenesis at criticality. *Proceedings of the National Academy of Sciences*, 111(10):3683–3688, 2014.

[72] Mark Land and Richard K Belew. No perfect two-state cellular automata for density classification exists. *Physical review letters*, 74(25):5148, 1995.

[73] Daniel B Larremore, Woodrow L Shew, and Juan G Restrepo. Predicting criticality and dynamic range in complex networks: effects of topology. *Physical review letters*, 106(5):058101, 2011.

[74] Reinhard Laubenbacher and Bernd Sturmfels. Computer algebra in systems biology. *American Mathematical Monthly*, 116(10):882–891, 2009.

[75] Lori Layne, Elena Dimitrova, and Matthew Macauley. Nested canalyzing depth and network stability. *Bulletin of mathematical biology*, 74(2):422–433, 2012.

[76] Saskia Le Cessie and Johannes C Van Houwelingen. Ridge estimators in logistic regression. *Applied statistics*, pages 191–201, 1992.

[77] Deok-Sun Lee and Heiko Rieger. Broad edge of chaos in strongly heterogeneous boolean networks. *Journal of Physics A: Mathematical and Theoretical*, 41(41):415001, 2008.

[78] Yuan Li, John O Adeyeye, and Reinhard Laubenbacher. Nested canalyzing functions and their average sensitivities. *arXiv preprint arXiv:1111.7217*, 2011.

[79] Min Liu and Kevin E Bassler. Emergent criticality from coevolution in random boolean networks. *Physical Review E*, 74(4):041910, 2006.

[80] Joseph Lizier, Mikhail Prokopenko, and Albert Zomaya. Detecting non-trivial computation in complex dynamics. *Advances in Artificial Life*, pages 895–904, 2007.

[81] Joseph T Lizier, Benjamin Flecker, and Paul L Williams. Towards a synergy-based approach to measuring information modification. In *Artificial Life (ALIFE), 2013 IEEE Symposium on*, pages 43–51. IEEE, 2013.

[82] Bartolo Luque and Ricard V Solé. Lyapunov exponents in random boolean networks. *Physica A: Statistical Mechanics and its Applications*, 284(1):33–45, 2000.

[83] Alex Madrahimov, Tomáš Helikar, Bryan Kowal, Guoqing Lu, and Jim Rogers. Dynamics of influenza virus and human host interactions during infection and replication cycle. *Bulletin of mathematical biology*, 75(6):988–1011, 2013.

[84] Zhongxing Mai and Haiyan Liu. Boolean network-based analysis of the apoptosis network: irreversible apoptosis and stable surviving. *Journal of Theoretical Biology*, 259(4):760–769, 2009.

[85] Santosh Manicka. Booleancanalization: source code for this dissertation. `https://github.com/SantoshManicka/BooleanCanalization`, 2016.

[86] Manuel Marques-Pita, Melanie Mitchell, and Luis M Rocha. The role of conceptual structure in designing cellular automata to perform collective computation. In *International Conference on Unconventional Computation*, pages 146–163. Springer, 2008.

[87] Manuel Marques-Pita and Luis M Rocha. Schema redescription in cellular automata: Revisiting emergence in complex systems. In *Artificial Life (ALIFE), 2011 IEEE Symposium on*, pages 233–240. IEEE, 2011.

[88] Manuel Marques-Pita and Luis M Rocha. Canalization and control in automata networks: body segmentation in drosophila melanogaster. *PloS one*, 8(3):e55946, 2013.

[89] Manuel Marques-Pita and Luis Mateus Rocha. Conceptual structure in cellular automata-the density classification task. In *ALIFE*, pages 390–397. Citeseer, 2008.

[90] Pablo Martínez-Sosa and Luis Mendoza. The regulatory network that controls the differentiation of t lymphocytes. *Biosystems*, 113(2):96–103, 2013.

[91] Edward J McCluskey. Minimization of boolean functions. *Bell system technical Journal*, 35(6):1417–1444, 1956.

[92] Scott Menard. *Logistic regression: From introductory to advanced concepts and applications.* Sage, 2010.

[93] Bratislav Mišić, Richard F Betzel, Azadeh Nematzadeh, Joaquin Goñi, Alessandra Griffa, Patric Hagmann, Alessandro Flammini, Yong-Yeol Ahn, and Olaf Sporns. Cooperative and competitive spreading dynamics on the human connectome. *Neuron*, 86(6):1518–1529, 2015.

[94] Melanie Mitchell, James P Crutchfield, and Peter T Hraber. Dynamics, computation, and the "edge of chaos": A re-examination. *arXiv preprint adap-org/9306003*, 1993.

[95] Melanie Mitchell, James P Crutchfield, and Peter T Hraber. Evolving cellular automata to perform computations: Mechanisms and impediments. *Physica D: Nonlinear Phenomena*, 75(1-3):361–391, 1994.

[96] Melanie Mitchell et al. Computation in cellular automata: A selected review. *Nonstandard Computation*, pages 95–140, 1996.

[97] Melanie Mitchell, Peter Hraber, and James P Crutchfield. Revisiting the edge of chaos: Evolving cellular automata to perform computations. *arXiv preprint adap-org/9303003*, 1993.

[98] Jacques Monod. On chance and necessity. In *Studies in the Philosophy of Biology*, pages 357–375. Springer, 1974.

[99] Thierry Mora and William Bialek. Are biological systems poised at criticality? *Journal of Statistical Physics*, 144(2):268–302, 2011.

[100] David Murrugarra, Alan Veliz-Cuba, Boris Aguilar, and Reinhard Laubenbacher. Identification of control targets in boolean molecular network models via computational algebra. *BMC Systems Biology*, 10(1):94, 2016.

[101] Azadeh Nematzadeh, Emilio Ferrara, Alessandro Flammini, and Yong-Yeol Ahn. Optimal network modularity for information diffusion. *Physical review letters*, 113(8):088701, 2014.

[102] Geoffrey R Norman and David L Streiner. *Biostatistics: the bare essentials.* PMPH-USA, 2008.

[103] Norman H Packard. *Adaptation toward the edge of chaos.* University of Illinois at Urbana-Champaign, Center for Complex Systems Research, 1988.

[104] Joshua L Payne, Jason H Moore, and Andreas Wagner. Robustness, evolvability, and the logic of genetic regulation. *Artificial life*, 20(1):111–126, 2014.

[105] G Barry Pierce and Carol Wallace. Differentiation of malignant to benign cells. *Cancer research*, 31(2):127–134, 1971.

[106] Andrew Pomerance, Edward Ott, Michelle Girvan, and Wolfgang Losert. The effect of network topology on the stability of discrete state models of genetic control. *Proceedings of the National Academy of Sciences*, 106(20):8209–8214, 2009.

[107] Christine Queitsch, Todd A Sangster, and Susan Lindquist. Hsp90 as a capacitor of phenotypic variation. *Nature*, 417(6889):618–624, 2002.

[108] CJ Olson Reichhardt and Kevin E Bassler. Canalization and symmetry in boolean models for genetic regulatory networks. *Journal of Physics A: Mathematical and Theoretical*, 40(16):4339, 2007.

[109] Juan G Restrepo, Edward Ott, and Brian R Hunt. Approximating the largest eigenvalue of network adjacency matrices. *Physical Review E*, 76(5):056119, 2007.

[110] Andrea Roli, Marco Villani, Alessandro Filisetti, and Roberto Serra. Dynamical criticality: overview and open questions. *arXiv preprint arXiv:1512.05259*, 2015.

[111] Suzanne L Rutherford and Susan Lindquist. Hsp90 as a capacitor for morphological evolution. *Nature*, 396(6709):336–342, 1998.

[112] Julio Saez-Rodriguez, Luca Simeoni, Jonathan A Lindquist, Rebecca Hemenway, Ursula Bommhardt, Boerge Arndt, Utz-Uwe Haus, Robert Weismantel, Ernst D Gilles, Steffen Klamt, et al. A logical model provides insights into t cell receptor signaling. *PLoS Comput Biol*, 3(8):e163, 2007.

[113] C Seshadhri, Yevgeniy Vorobeychik, Jackson R Mayo, Robert C Armstrong, and Joseph R Ruthruff. Influence and dynamic behavior in random boolean networks. *Physical review letters*, 107(10):108701, 2011.

[114] Cosma Shalizi. *Advanced data analysis from an elementary point of view.* Citeseer, 2013.

[115] Claude E Shannon. A symbolic analysis of relay and switching circuits. *Electrical Engineering*, 57(12):713–723, 1938.

[116] Ilya Shmulevich, Edward R Dougherty, and Wei Zhang. Gene perturbation and intervention in probabilistic boolean networks. *Bioinformatics*, 18(10):1319–1331, 2002.

[117] Ilya Shmulevich and Stuart A Kauffman. Activities and sensitivities in boolean network models. *Physical review letters*, 93(4):048701, 2004.

[118] Ilya Shmulevich, Stuart A Kauffman, and Maximino Aldana. Eukaryotic cells are dynamically ordered or critical but not chaotic. *Proceedings of the National Academy of Sciences of the United States of America*, 102(38):13439–13444, 2005.

[119] Martin Slawski, Martin Daumer, and Anne-Laure Boulesteix. Cma–a comprehensive bioconductor package for supervised classification with high dimensional data. *BMC bioinformatics*, 9(1):439, 2008.

[120] David Slepian. On the number of symmetry types of boolean functions of n variables. *Canad. J. Math*, 5(2):185–193, 1953.

[121] Shane Squires, Edward Ott, and Michelle Girvan. Dynamical instability in boolean networks as a percolation problem. *Physical review letters*, 109(8):085701, 2012.

[122] Shane Squires, Andrew Pomerance, Michelle Girvan, and Edward Ott. Stability of boolean networks: The joint effects of topology and update rules. *Physical Review E*, 90(2):022814, 2014.

[123] Stephen C Stearns. Progress on canalization. *Proceedings of the National Academy of Sciences*, 99(16):10229–10230, 2002.

[124] Steven H Strogatz. *Nonlinear dynamics and chaos: with applications to physics, biology, chemistry, and engineering.* Westview press, 2014.

[125] Kristóf Z Szalay and Peter Csermely. Perturbation centrality and turbine: a novel centrality measure obtained using a versatile network dynamics tool. *PloS one*, 8(10):e78059, 2013.

[126] René Thomas. *Kinetic Logic: A Boolean Approach to the Analysis of Complex Regulatory Systems: Proceedings of the EMBO Course "Formal Analysis of Genetic Regulation", Held in Brussels, September 6–16, 1977*, volume 29. Springer Science & Business Media, 2013.

[127] Sergi Valverde, Sebastian Ohse, Malgorzata Turalska, Bruce J West, and Jordi Garcia-Ojalvo. Structural determinants of criticality in biological networks. *Frontiers in physiology*, 6, 2015.

[128] Alan Veliz-Cuba and Brandilyn Stigler. Boolean models can explain bistability in the lac operon. *Journal of computational biology*, 18(6):783–794, 2011.

[129] Rui-Sheng Wang and Réka Albert. Elementary signaling modes predict the essentiality of signal transduction network components. *BMC systems biology*, 5(1):1, 2011.

[130] Rui-Sheng Wang, Bradley A Maron, and Joseph Loscalzo. Systems medicine: evolution of systems biology from bench to bedside. *Wiley Interdisciplinary Reviews: Systems Biology and Medicine*, 7(4):141–161, 2015.

[131] Wikipedia. Evaluation of binary classifiers — Wikipedia, the free encyclopedia. `https://en.wikipedia.org/wiki/Evaluation_of_binary_classifiers`, 2016. [Online; accessed 26-Sep-2016].

[132] Wikipedia. Hyperoctahedral group — Wikipedia, the free encyclopedia. `https://en.wikipedia.org/wiki/Hyperoctahedral_group`, 2016. [Online; accessed 30-Jan-2017].

[133] Wikipedia. Iterated function — Wikipedia, the free encyclopedia. `https://en.wikipedia.org/wiki/Iterated_function`, 2016. [Online; accessed 9-Mar-2017].

[134] Wikipedia. Minkowski–bouligand dimension — Wikipedia, the free encyclopedia. `https://en.wikipedia.org/wiki/Minkowski%E2%80%93Bouligand_dimension`, 2016. [Online; accessed 15-Feb-2017].

[135] Wikipedia. Receiver operating characteristic — Wikipedia, the free encyclopedia. `https://en.wikipedia.org/wiki/Receiver_operating_characteristic`, 2016. [Online; accessed 13-Jan-2017].

[136] Kai Willadsen and Janet Wiles. Robustness and state-space structure of boolean gene regulatory models. *Journal of theoretical biology*, 249(4):749–765, 2007.

[137] Stephen Wolfram et al. *Theory and applications of cellular automata*, volume 1. World scientific Singapore, 1986.

[138] Andrew Wuensche. The ghost in the machine. In *Santa Fe Institute Studies In The Sciences Of Complexity-Proceedings*, volume 17, pages 465–465. Addison-Wesley Publishing Co, 1994.

[139] Gang Yang, Colin Campbell, and Réka Albert. Compensatory interactions to stabilize multiple steady states or mitigate the effects of multiple deregulations in biological networks. *Physical Review E*, 94(6):062316, 2016.

[140] Jorge GT Zañudo and Réka Albert. Cell fate reprogramming by control of intracellular network dynamics. *PLoS Comput Biol*, 11(4):e1004193, 2015.

[141] Ranran Zhang, Mithun Vinod Shah, Jun Yang, Susan B Nyland, Xin Liu, Jong K Yun, Réka Albert, and Thomas P Loughran. Network model of survival signaling in large granular lymphocyte leukemia. *Proceedings of the National Academy of Sciences*, 105(42):16308–16313, 2008.

Appendix A

Calculating k_u

In this appendix, we describe a method to calculate the k_u of a Boolean automaton (Boolean function).

A.1 Concepts

The central concepts required to understand the formulation of k_u are defined in Chapters 1 and 3. Below, we introduce additional concepts necessary to understand the procedure for calculating k_u.

Cubes and subcubes. As described in Chapter 1, every k-input Boolean function can be represented as a k-dimensional hypercube or simply *cube*. Every such cube contains *subcubes* of every possible dimension from k down to 0 — the entire cube is the only subcube of dimension k, and every individual input vector (corner) is a subcube of dimension 0. Every subcube can be uniquely identified by an *identifier* that consists of input variables and their values that remain constant in the subcube. For example, consider the following 2-dimensional subcube consisting of the following input vectors in a 3-dimensional cube: $\{(0,1,0),(0,1,1),(1,1,0),(1,1,1)\}$. In this subcube, the only input variable whose value is a constant is i_2 with value 1. Thus, the identifier of this subcube is: $(i_2 = 1)$. A subcube identifier could involve multiple input variables, e.g., $(i_1 = 0, i_3 = 1, i_4 = 0)$.

As a special case, the identifier of an input vector is that vector itself. A set of *parallel subcubes* consists of subcubes whose identifiers contain the same set of input variables; each subcube is distinguished by the values of the identifier variables. The following is a single set of parallel subcubes, for example: $\{(i_1 = 0, i_3 = 0), (i_1 = 0, i_3 = 1), (i_1 = 1, i_3 = 0), (i_1 = 1, i_3 = 1)\}$. The dimension of a subcube is equal to k minus the number of its identifier variables. In the parallel subcube above, $k = 4$ means that then dimension of every subcube in the set is equal to 2. For a cube of dimension k, the number of possible sets of parallel subcubes of dimension D is equal to $C(k, D)$. For example, in a $k = 3$ cube, the number of sets of parallel subcubes of dimension $D = 2$ is equal to $C(3, 2) = 3$ — the 3 pairs of parallel faces (left-right, front-back, top-bottom) of the cube.

Composite schemata. A *composite schema* is a two-symbol schema with at least one fixed (non-permuting) '#' and at least one permuting permuting '#'; it is named so because it contains the characteristic features of both a wildcard schema and a two-symbol schema. For example, $(1, \#, \dot{0}, \dot{\#})$ is a composite schema, whereas $(1, 0, \dot{0}, \dot{\#})$ or $(1, 0, \dot{\#}, \dot{\#})$ is an ordinary two-symbol schema. One way to identify a composite schema is by combining two or more two-symbol schemata from a set of parallel subcubes. For example, $(\#, \dot{0}, \dot{1}, \dot{\#})$ can be obtained by combining $(0, \dot{0}, \dot{1}, \dot{\#})$ and $(1, \dot{0}, \dot{1}, \dot{\#})$, or by combining $(0, \dot{0}, \dot{\#}, \dot{\#})$ and $(1, \dot{1}, \dot{\#}, \dot{\#})$. Notice in the former that the composite schema is a full union of the combining schemata, whereas in the latter it is a union of portions of the combining schemata. In either case, the composite schema acts as "bridge" that unites ordinary parallel two-symbol schemata—this is essentially why we consider parallel subcubes. A set of two-symbol schemata can combine to more than one composite schemata. For example, the following set of two-symbol schemata $\{(0, 0, \dot{0}, \dot{\#}), (0, 1, \dot{0}, \dot{\#}), (1, 0, \dot{0}, \dot{\#})\}$ combine to produce the following set of composite schemata: $\{(0, \#, \dot{0}, \dot{\#}), (\#, 0, \dot{0}, \dot{\#})\}$; if the original set was rather $\{(0, 1, \dot{0}, \dot{\#}), (1, 0, \dot{0}, \dot{\#})\}$, no composite schemata would be possible since the values of subcube identifier variables (i_1 and i_2) can't combine.

A.2 Method

The procedure for calculating k_u involves the following steps:

Step 0: Initialize a "cover list" of length 2^k with all zeros — this list shall contain the dimensions of the largest covering two-symbol schemata corresponding to each input vector, and will be updated throughout the procedure.

Repeat steps 1 to 4 below for every possible dimension D in decreasing order from k to 1, and for every possible set of parallel cubes of a given dimension.

Step 1: Consider a single set of parallel subcubes of a given dimension.

Step 2: Identify all ordinary two-symbol schemata in each subcube in the set obtained in the previous step and for each output value.

Note that a subcube may contain more than one two-symbol schema. Every two-symbol schema in a subcube of dimension D must contain exactly D permuting symbols. One way to identify a two-symbol schema is to compute the number of input vectors in the subcube that contain a certain number of 1s (n_1). If that number is equal to $C(D, n_1)$, then those input vectors may constitute a two-symbol schema. Note that we identify only those two-symbol schemata that cover input vectors with the associated number of 1s in some interval $[n_1, n_1 + w]$ where $w \geq 1$.

Step 3: Identify all composite schemata from the set of ordinary two-symbol schemata obtained in the previous step and for each output value.

One way to identify a composite schema is to enumerate "signatures" of all possible ordinary two-symbol schemata of dimension D and match them against the permuting symbols of the schemata obtained in the previous step. If more than one two-symbol schema in the set matches a given signature, then it is an indication that they might combine to form a composite schema. An important point to note here is that a signature may only *partially* match the schemata and yet produce a composite schema. A representative example is: the signature $(\dot{0}, \dot{1}, \dot{\#})$ partially matches the permuting symbols of both the two-symbol schemata $(0, \dot{0}, \dot{\#}, \dot{\#})$ and $(1, \dot{1}, \dot{\#}, \dot{\#})$, to produce the composite schema $(\#, \dot{0}, \dot{1}, \dot{\#})$.

Step 4: Record the dimension of the two-symbol schema against every input vector it covers if and only if the current largest covering two-symbol schema's dimension is smaller.

Finally, compute k_r^* by averaging over the covering dimensions of all 2^k input vectors. Compute $k_u = k - k_r^*$.

A.3 Example

In this section, we apply the procedure described above to calculate the k_u of an example $k = 3$ function (Fig. A.1), and describe the steps involved in detail.

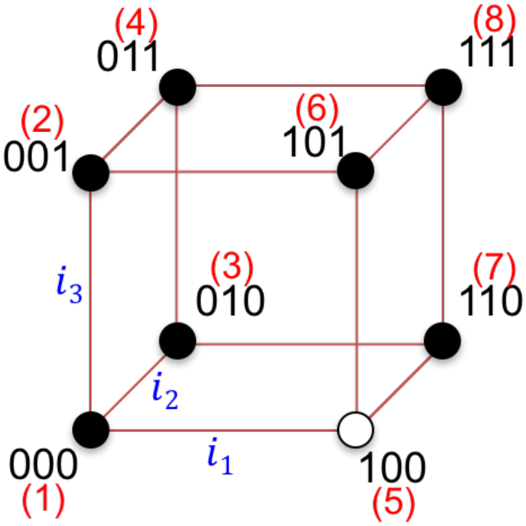


Figure A.1: An example $k = 3$ Boolean function $x = f(i_1, i_2, i_3)$. Each corner (input vector) is marked with a decimal number in red — to be used as an index to refer to the input vectors in the cover list.

Initialize cover list: $(0, 0, 0, 0, 0, 0, 0, 0)$, where the order is the same as the numbers indicated in red in the figure.

D = 3:

Step 1. Since $D = k$, the only subcube in the set of parallel subcubes is the full cube itself.

Step 2. The only two-symbol schema in this subcube is: $\{(\dot{1}, \dot{1}, \dot{\#})\}$, corresponding to output 1.

Step 3. No composite schemata exist since there is only two-symbol schema.

Step 4. The set of all two-symbol schemata now consists of $\{(\dot{1}, \dot{1}, \dot{\#})\}$, with dimension equal to $\log_2(4) = 2$, and covering the corners {(4,6,7,8)}. The updated cover list is: (0,0,0,2,0,2,2,2).

D = 2:

There are 3 sets of parallel subcubes: $\{(i_1 = 0), (i_1 = 1)\}$, $\{(i_2 = 0), (i_2 = 1)\}$ and $\{(i_3 = 0), (i_3 = 1)\}$.

Step 1a. Consider the set of parallel subcubes $\{(i_1 = 0), (i_1 = 1)\}$.

Step 2a. The set of two-symbol schema in this set of parallel subcubes consists of: $\{(0, \dot{\#}, \dot{\#})$, $(1, \dot{1}, \dot{\#})\}$, corresponding to output 1.

Step 3a. The set of composite schemata obtained by combining the two-symbol schemata in the set from above consists of: $\{(\#, \dot{1}, \dot{\#})\}$.

Step 4a. The set of all two-symbol schemata now consists of $\{(0, \dot{\#}, \dot{\#}), (1, \dot{1}, \dot{\#}), (\#, \dot{1}, \dot{\#})\}$, with dimensions respectively equal to $\log_2(4) = 2$, $\log_2(3) \approx 1.58$ and $\log_2(6) \approx 2.58$, and covering the corners {(1,2,3,4),(6,7,8),(2,3,4,6,7,8)} respectively. The updated cover list is: (2,2.58,2.58,2.58,0,2.58,2.58,2.58).

Step 1b. Consider the set of parallel subcubes $\{(i_2 = 0), (i_2 = 1)\}$.

Step 2b. The set of two-symbol schema in this set of parallel subcubes consists of: $\{(\dot{\#}, 1, \dot{\#})\}$, corresponding to output 1.

Step 3b. No composite schemata exist since there is only two-symbol schema.

Step 4b. The set of all two-symbol schemata now consists of $\{(\dot{\#}, 1, \dot{\#})\}$, with a dimensions equal to $\log_2(4) = 2$, and covering the corners {(3,4,7,8)}. The updated cover list is: (2,2.58,2.58,2.58,0,2.58,2.58,2.58) (no alteration).

Step 1c. Consider the set of parallel subcubes $\{(i_3 = 0), (i_3 = 1)\}$.

Step 2c. The set of two-symbol schema in this set of parallel subcubes consists of: $\{(\dot{\#}, \dot{\#}, 1)\}$, corresponding to output 1.

Step 3c. No composite schemata exist since there is only two-symbol schema.

Step 4c. The set of all two-symbol schemata now consists of $\{(\dot{\#}, \dot{\#}, 1)\}$, with a dimension equal to $\log_2(4) = 2$, and covering the corners {(2,4,6,8)}. The updated cover list is: (2,2.58,2.58,2.58,0,2.58,2.58,2.58) (no alteration).

D = 1:

There are 3 sets of parallel subcubes: $\{(i_1 = 0, i_2 = 0), (i_1 = 0, i_2 = 1), (i_1 = 1, i_2 = 0), (i_1 = 1, i_2 = 1)\}$, $\{(i_1 = 0, i_3 = 0), (i_1 = 0, i_3 = 1), (i_1 = 1, i_3 = 0), (i_1 = 1, i_3 = 1)\}$ and $\{(i_2 = 0, i_3 = 0), (i_2 = 0, i_3 = 1), (i_2 = 1, i_3 = 0), (i_2 = 1, i_3 = 1)\}$.

Step 1a. Consider the set of parallel subcubes $\{(i_1 = 0, i_2 = 0), (i_1 = 0, i_2 = 1), (i_1 = 1, i_2 = 0), (i_1 = 1, i_2 = 1)\}$.

Step 2a. The set of two-symbol schema in this set of parallel subcubes consists of: $\{(0, 0, \dot{\#}),$

$(0,1,\dot{\#}),(1,1,\dot{\#}\}$, corresponding to output 1.

Step 3a. No composite schemata exist. Note that $(0,0,\dot{\#})$ and $(0,1,\dot{\#})$ can combine to form $(0,\dot{\#},\dot{\#})$ but it is not a valid composite schema (see definition in Sec. A.1).

Step 4a. The set of all two-symbol schemata now consists of $\{(0,0,\dot{\#}),(0,1,\dot{\#}),(1,1,\dot{\#}\}$, with dimensions respectively equal to $\log_2(2)=1$ each, and covering the corners {(1,2),(3,4),(7,8)} respectively. The updated cover list is: (2,2.58,2.58,2.58,0,2.58,2.58,2.58)s (no alteration).

Step 1b. Consider the set of parallel subcubes $\{(i_1=0,i_3=0),(i_1=0,i_3=1),(i_1=1,i_3=0),(i_1=1,i_3=1)\}$.

Step 2b. The set of two-symbol schema in this set of parallel subcubes consists of: $\{(0,\dot{\#},0),(0,\dot{\#},1),(1,\dot{\#},1)\}$, corresponding to output 1.

Step 3b. No composite schemata exist.

Step 4b. The set of all two-symbol schemata now consists of $\{(0,\dot{\#},0),(0,\dot{\#},1),(1,\dot{\#},1\}$, with dimensions respectively equal to $\log_2(2)=1$ each, and covering the corners {(1,3),(2,4),(6,8)} respectivelu. The updated cover list is: (2,2.58,2.58,2.58,0,2.58,2.58,2.58) (no alteration).

Step 1c. Consider the set of parallel subcubes $\{(i_2=0,i_3=0),(i_2=0,i_3=1),(i_2=1,i_3=0),(i_2=1,i_3=1)\}$.

Step 2c. The set of two-symbol schema in this set of parallel subcubes consists of: $\{(\dot{\#},0,1),(\dot{\#},1,1),(\dot{\#},1,0)\}$, corresponding to output 1.

Step 3c. No composite schemata exist.

Step 4c. The set of all two-symbol schemata now consists of $\{(\dot{\#},0,1),(\dot{\#},1,1),(\dot{\#},1,0)\}$, with dimensions respectively equal to $\log_2(2)=1$ each, and covering the corners {(2,6),(4,8),(3,7)} respectively. The updated cover list is: (2,2.58,2.58,2.58,0,2.58,2.58,2.58) (no alteration).

Finally, k_r^* is the mean of the values in the cover list: $k_r^* = 2.185 \implies k_u = k - k_r^* = 0.815$ (exact value is 0.8112781 if the log values above are not rounded).

A.4 Source code

An implementation of the above is available in **R**; the link to the *GitHub* repository is listed in Ref. [85]. The main files are:

1. *ComputeKu.R*: The main file containing an implementation of the procedure to compute k_u.
2. *ComputeDetectCubes.R*: A supporting file containing an implementation of a part of Step 3 of the procedure that helps identify composite schemata. Specifically, it helps identify the non-permuting wildcard symbols in a two-symbol schema.

Appendix B

Integrating a Boolean network

In this appendix, we describe a method to integrate a Boolean network.

B.1 Concepts

The central concepts and notations required to understand the integration procedure are described in Chapter 4. Below, we introduce additional concepts necessary to understand the details of the procedure.

Sets of schemata. A 'set' of schemata is defined as a set where the logical condition specified by at least of one of the schemata is true. In other words, a set of schemata specifies a logical condition in the form of a disjunction of conjunctive clauses. A set of schemata naturally redescribes a set of LUT entries. For example, $\{10\#, \#\#1\}$ specifies the logical condition: $(x_1 = 1 \wedge x_2 = 0) \vee (x_3 = 1)$, and redescribes the set of LUT entries $\{(1, 0, 0), (1, 0, 1), (0, 0, 1), (0, 1, 1), (1, 1, 1)\}$.

Union of a set of schemata. 'Union' is defined as a unary operation on a set of schemata that returns the set of *all* possible *minimal* schemata which jointly cover all of the LUT entries that the original set covers. In other words, the union of a set of implicants or prime implicants is the set of *all* prime implicants that covers the same set of LUT entries that the original set does. The result of an union could comprise more or fewer schemata than the original set depending on its composition. For example, the union of the set {00#, 01#} is the set {0##}; whereas, the union of {0#0, 11#} is the set {0#0, #10, 11#}. The union operation is equivalent to converting all the schemata in the original set into LUT entries first and then compressing it using a standard logic minimization method such as Quine-McCluskey to obtain the set of all prime implicants. This latter procedure is clearly inefficient since it involves "decompression" first followed by a compression from scratch; utilizing the compression that already comes with the original set of schemata would be more efficient. In fact, the union operation could be thought of as a nonlinear extension of Quine-McCluskey because, in the latter smaller schemata (fewer wildcards) combine to form only larger schemata, whereas a union of larger schemata could result in smaller schemata as well. For example, the union of the set {00#0#0, ##111#} is the set {00#0#0, ##111#, 001#10} (notice the additional schema with just one wildcard).

Intersection of sets of schemata. 'Intersection' is defined as a binary operation on a pair of sets of schemata that returns a single set of *all* possible *minimal* schemata which jointly cover all of the LUT entries common to both sets. In other words, the intersection of a pair of sets of implicants or prime implicants is the set of all *all* prime implicants that covers the set of all LUT entries that both the intersecting sets cover. Thus, the intersection operation is nothing but an implementation of the distributive law of Boolean algebra [28]. The result of an intersection is the empty set {ϕ} if the intersecting sets have no LUT entries in common. Here are a few examples: {#1#} intersection {1##} = {11#}; {##1} intersection {11#, #11} = {#11}; {1##, #00} intersection {#11, 11#} = {11#}; {##1} intersection {10#, #11} = {1#1, #11}; {1##, ##1} intersection {#1#} = {11#, #11}; {0#} intersection {1#} = {ϕ}; and {0#} intersection {1#, #1} = {01}. Naturally, any number of sets of schemata can be intersected by intersecting the first pair of sets, then replacing the pair in the original set with their intersection, and continue so on until a single (potentially empty) set of schemata remains.

B.2 Method

The procedure for integrating a BN involves the following steps:

For each node and for each output (0 and 1), repeat the following steps for a specified number of integration steps:

Step 1: Compute the set of predecessor schemata for each schema mapping to the given output.

The set of predecessor schemata of a given schema is the intersection of the sets of predecessor schemata of the atomic schemata associated with its individual literals.

Step 2: Compress the set of all predecessor schemata sets obtained in step 1.

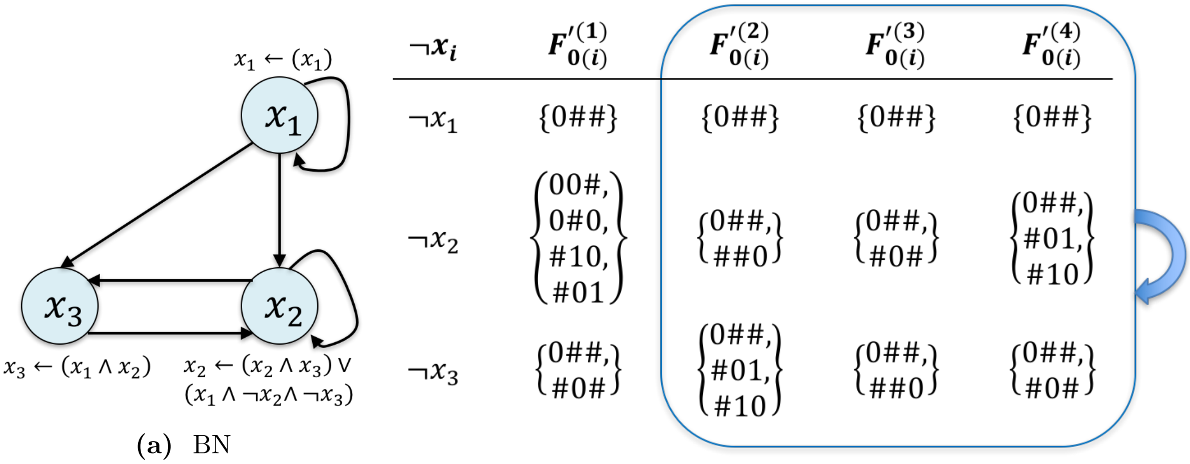
This set is just the union of the set of all predecessor schemata sets obtained in step 1.

B.3 Example

In this section, we apply the procedure described above to integrate the example BN described in Chapter 4, and describe the steps involved in detail. For clarity, we repeat the BN and the associated Boolean function sequence (BFS) for output 0 in Fig.B.1.

For simplicity, we only describe the steps involved in the first step of integration for the atomic schemata containing a '0'. That is, we only show how to compute $F_0'^{(2)}$. Note that $F_0'^{(1)}$ comprises the set of input schemata of each node corresponding to output 0, which we list below (note that all schemata have $n = 3$ symbols; symbols corresponding to nodes that are not inputs are just wildcards):

1. Node x_1: $\{0\#\#\} \mapsto 0$; $\{1\#\#\} \mapsto 1$.



(a) BN

$\neg x_i$	$F'^{(1)}_{0(i)}$	$F'^{(2)}_{0(i)}$	$F'^{(3)}_{0(i)}$	$F'^{(4)}_{0(i)}$
$\neg x_1$	{0##}	{0##}	{0##}	{0##}
$\neg x_2$	{00#, 0#0, #10, #01}	{0##, ##0}	{0##, #0#}	{0##, #01, #10}
$\neg x_3$	{0##, #0#}	{0##, #01, #10}	{0##, ##0}	{0##, #0#}

(b) BFS

Figure B.1: An example BN and the BFS corresponding to atomic schemata containing a '0'.

2. Node x_2: $\{00\#, 0\#0, \#10, \#01\} \mapsto 0$; $\{100, \#11\} \mapsto 1$.

3. Node x_3: $\{0\#\#, \#0\#\} \mapsto 0$; $\{11\#\} \mapsto 1$.

We now proceed to calculating $F_0'^{(2)}$.

Node = x_1, output = 0:

Step 1. Compute the set of predecessor schemata for each schema in the set $\{0\#\#\}$ and intersect the resulting sets.

> Step 1a. Compute the predecessor schemata of $\{0\#\#\}$. First, retrieve the predecessor schemata of each literal in the schema: predecessor schemata of $x_1 = 0$: $\{0\#\#\}$. Intersection is not necessary since there are no more predecessor schemata to retrieve. Therefore, the result of intersection is: $\{0\#\#\}$.

Step 2. Compress the set of all predecessor schemata obtained in step 1: $\{0\#\#\}$. Since there is only one schema in the set, the result of union is: $\{0\#\#\}$.

Therefore, $F_0'^{(2)}$ for node x_1 is: $\{0\#\#\}$.

Node = x_2, output = 0:

Step 1. Compute the set of predecessor schemata for each schema in the set $\{00\#, 0\#0, \#10, \#01\}$ and intersect the resulting sets.

Step 1a. Compute the predecessor schemata of $\{00\#\}$. First, retrieve the predecessor schemata of each literal in the schema: predecessor schemata of $x_1 = 0$: $\{0\#\#\}$; predecessor schemata of $x_2 = 0$: $\{00\#, 0\#0, \#10, \#01\}$. Then, intersect the sets $\{0\#\#\}$ and $\{00\#, 0\#0, \#10, \#01\}$. The result of intersection is: $\{00\#, 0\#0\}$.

Step 1b. Compute the predecessor schemata of $\{0\#0\}$. First, retrieve the predecessor schemata of each literal in the schema: predecessor schemata of $x_1 = 0$: $\{0\#\#\}$; predecessor schemata of $x_3 = 0$: $\{0\#\#, \#0\#\}$. Then, intersect the sets $\{0\#\#\}$ and $\{0\#\#, \#0\#\}$. The result of intersection is: $\{0\#\#\}$.

Step 1c. Compute the predecessor schemata of $\{\#10\}$. First, retrieve the predecessor schemata of each literal in the schema: predecessor schemata of $x_2 = 1$: $\{100, \#11\}$; predecessor schemata of $x_3 = 0$: $\{0\#\#, \#0\#\}$. Then, intersect the sets $\{100, \#11\}$ and $\{0\#\#, \#0\#\}$. The result of intersection is: $\{100, 011\}$.

Step 1d. Compute the predecessor schemata of $\{\#01\}$. First, retrieve the predecessor schemata of each literal in the schema: predecessor schemata of $x_2 = 0$: $\{00\#, 0\#0, \#10, \#01\}$; predecessor schemata of $x_3 = 1$: $\{11\#\}$. Then, intersect the sets $\{00\#, 0\#0, \#10, \#01\}$ and $\{11\#\}$. The result of intersection is: $\{110\}$.

Step 2. Compress the set of all predecessor schemata obtained in step 1: $\{00\#, 0\#0, 0\#\#, 100, 011, 110\}$. The result of the union is: $\{\#\#0, 0\#\#\}$.

Therefore, $F_0^{\prime(2)}$ for node x_2 is: $\{\#\#0, 0\#\#\}$.

Node = x_3, output = 0:

Step 1. Compute the set of predecessor schemata for each schema in the set $\{0\#\#, \#0\#\}$ and intersect the resulting sets.

> Step 1a. Compute the predecessor schemata of $\{0\#\#\}$. First, retrieve the predecessor schemata of each literal in the schema: predecessor schemata of $x_1 = 0$: $\{0\#\#\}$. Intersection is not necessary since there are no more predecessor schemata to retrieve. Therefore, the result of intersection is: $\{0\#\#\}$.
>
> Step 1b. Compute the predecessor schemata of $\{\#0\#\}$. First, retrieve the predecessor schemata of each literal in the schema: predecessor schemata of $x_2 = 0$: $\{00\#, 0\#0, \#10, \#01\}$. Intersection is not necessary since there are no more predecessor schemata to retrieve. Therefore, the result of intersection is: $\{00\#, 0\#0, \#10, \#01\}$.

Step 2. Compress the set of all predecessor schemata obtained in step 1: $\{0\#\#, 00\#, 0\#0, \#10, \#01\}$. The result of the union is: $\{0\#\#, \#10, \#01\}$.

Therefore, $F_0'^{(2)}$ for node x_3 is: $\{0\#\#, \#10, \#01\}$.

This completes the computation of $F_0'^{(2)}$ for all three nodes in the BN.

B.4 Source code

An implementation of the above is available in **R**; the link to the *GitHub* repository is listed in Ref. [85]. The main files are:

1. *ComputeIntegrateBoolNet.R*: The main file containing an implementation of the Boolean network integration procedure.

2. *ComputeSchemaSetOperations.R*: A supporting file containing implementations of the union and intersection procedures.
3. *ComputeIntegratedKeff.R*: Contains an implementation of computing k_e of $F'^{(t)}$ using the output of the integration procedure.

Appendix C

Further properties of F''

The defining characteristics of F'' is described § 3.1. In this appendix, we highlight other important properties regarding the composition of F''.

1. **All wildcard schemata present in F' are present in F'' as well unless redescribed by two-symbol schemata containing at least as many wildcard symbols as the redescribed wildcard schemata.**

   The first part of that statement is a consequence of the defining characteristics of F'' described in § 3.1, which state that F'' contains all possible *maximal* two-symbol schemata; a wildcard schema is maximal if it cannot be redescribed by another two-symbol schema in F''. Here, we claim that a two-symbol schema with fewer wildcard symbols (permuting or otherwise) cannot redescribe a wildcard schema with more wildcard symbols. This is not a trivial claim since a two-symbol schema with fewer wildcards could redescribe a larger set of LUT entries than a wildcard schema with more wildcard symbols. For example, $(\dot{0}, \dot{1}, \dot{\#})$ redescribes 6 LUT entries, whereas $(1, \#, \#)$ redescribes only 4 LUT entries.

   A general way of stating that a two-symbol schema f''_θ redescribes a wildcard schema f'_υ is by implying that the set of redescribed LUT entries of the former Θ_θ is a superset of the redescribed LUT entries Υ_υ of the latter (§ 1.6.2). For example, $(\dot{1}, \dot{\#}, \dot{\#})$ redescribes $(1, 1, \#)$.

Following is an outline of the proof of the aforementioned claim.

Let f'_v be some wildcard schema that contains l_1 literals and w_1 wildcards, and let f''_θ be some two-symbol that contains l_2 literals and w_2 wildcards, with a total of d_2 fixed symbols and g_2 permuting symbols, such that $l_2 > l_1$ and $w_2 < w_1$, and $l_1 + w_1 = l_2 + w_2 = f_2 + g_2 = k$. That is, f''_θ contains fewer wildcards and more literals than f'_v.

A necessary but not sufficient condition for f''_θ to redescribe f'_v is that there exists a two-symbol schema f''_x that can be "extracted" from f''_θ such that the l_1 literals of f'_v match the symbols at the corresponding locations of f''_x and the d_2 fixed symbols of f''_x match the symbols at the corresponding locations of f'_v. For example, suppose that $f'_v = (\#, 1, 0, \#, \#)$ and $f''_\theta = (\#, 1, \dot{0}, \dot{1}, \dot{\#})$. Then, $f''_x = (\#, 1, 0, \dot{1}, \dot{\#})$, which is the first two-symbol schema listed in the following decomposition of f''_θ: $(\#, 1, \dot{0}, \dot{1}, \dot{\#}) = (\#, 1, 0, \dot{1}, \dot{\#}) \cup (\#, 1, 1, \dot{0}, \dot{\#}) \cup (\#, 1, \#, \dot{1}, \dot{0})$. In particular, this means that the condition part (§ 1.2) of f'_v matches the condition part of f''_x, implying that it is only the f''_x part of f''_θ that could potentially redescribe f'_v. We can now focus on f''_x, and in particular on the portion (or projection) of f''_x that contains the permuting symbols, denoted f''^p_x, and compare the corresponding portion of f'_v, denoted f'^p_v, since all other symbols in f''_x and f'_v match. Let m denote the number of symbols in f''^p_x and f'^p_v (they must contain the same number of symbols). By implication of the fact that $w_2 < w_1$ and the way by which f''_x is constructed, (i) f''^p_x contains at least one non-wildcard symbol; (ii) f'^p_v contains only wildcards. In the example above, $f''^p_x = (\dot{1}, \dot{\#})$, and $f'^p_v = (\#, \#)$. Clearly, f''^p_x does not redescribe f'^p_v since the latter redescribes exactly 2^m LUT entries, whereas the former redescribes fewer than 2^m LUT entries. This implies that f''_θ does not redescribe f'_v.

2. **F'' may contain more schemata than F' which are not essential for describing the automaton but are essential for capturing input symmetry that contributes to k_u. In particular, the additional two-symbol schemata may contain fewer wildcards than any other wildcard schema.**

Even though a wildcard schema with more wildcards cannot be *fully* redescribed by a two-symbol schema with fewer wildcards, it is possible that a subset of its redescribed LUT entries is redescribed by such a two-symbol schema that, moreover, is the one that contributes to k_u, with respect to those LUT entries. Here, we describe an example.

Consider a $k = 6$ automaton with the corresponding F_1' containing wildcard schemata with 2 wildcards each (Fig. C.1). The corresponding F_1'' contains a two-symbol schema, denoted f_d'' (Fig. C.1), that contains only one wildcard. This is a valid two-symbol schema because it is not fully redescribed by any other schema in F_1'', satisfying the defining characteristics of F'' (§ 3.1). Clearly, f_d'' is not necessary to describe the automaton since every one of its redescribed LUT entries is redescribed by some wildcard schema in F_1''. However, f_d'' contributes to k_u because its dimension is equal to $\log_2(6) \approx 2.58$ which is greater than the dimension of any of the wildcard schemata (equal to 2). Thus, the amount contributed to k_u by the LUT entries redescribed by f_d'' (e.g., $(0, 0, 1, 0, 1, 0)$) is about $k - 2.58 = 3.42$ and not $k - 2 = 4$. Thus, we say that f_d'' is an essential two-symbol schema as far as calculating k_u is concerned, as the stated goal of k_u is to capture all possible valid wildcard schemata and two-symbol schemata with the largest possible dimensions (§ 3.1).

F_1'

id	Schema
f_a'	(0,0, #, 0, #, 1)
f_b'	(#, 0,1, #, 1,0)
f_c'	(0, #, 1,1,0, #)

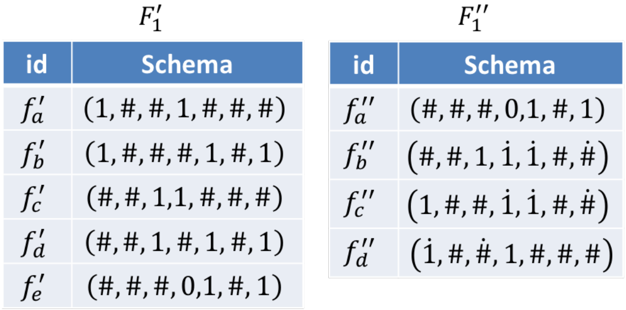
F_1''

id	Schema
f_a''	(0,0, #, 0, #, 1)
f_b''	(#, 0,1, #, 1,0)
f_c''	(0, #, 1,1,0, #)
f_d''	$(0,0,1, \dot{0}, \dot{1}, \dot{\#})$

Figure C.1: An example of a LUT where F_1'' contains more schemata than F_1', and the only two-symbol schema contains fewer wildcards than any wildcard schema.

As a second example, consider the 'GKL' rule discussed in Chapter 4 (Fig. 4.14). The corresponding sets F_1' and F_1'' are reproduced here for clarity (Fig. C.2). Here, the schemata f_b''

and f''_c in F''_1 are not essential to describe the automaton, because the set of essential wildcard schemata consists of f'_a, f'_c and f'_e, which are redescribed by just f''_a and f''_d. However, f''_b and f''_c are valid two-symbol schemata since they are not fully redescribed by any other two-symbol schemata in F''_1, satisfying the defining characteristics of F'' (§ 3.1). More importantly, they have a dimension equal to $\log_2(32) = 5$ which is greater than the dimension of f''_a equal to $\log_2(16) = 4$. Therefore, all LUT entries shared between f''_a and f''_b (e.g., $(0, 0, 1, 0, 1, 0, 1)$), or between f''_a and f''_c (e.g., $(1, 0, 0, 0, 1, 0, 1)$) contribute a value of $k - 5 = 2$ to k_u, and not $k - 4 = 3$. Thus, f''_b and f''_c are essential for capturing input symmetry, thus contributing to k_u, even though they are not essential to describe the automaton itself.



F'_1

id	Schema
f'_a	$(1, \#, \#, 1, \#, \#, \#)$
f'_b	$(1, \#, \#, \#, 1, \#, 1)$
f'_c	$(\#, \#, 1, 1, \#, \#, \#)$
f'_d	$(\#, \#, 1, \#, 1, \#, 1)$
f'_e	$(\#, \#, \#, 0, 1, \#, 1)$

F''_1

id	Schema
f''_a	$(\#, \#, \#, 0, 1, \#, 1)$
f''_b	$(\#, \#, 1, \dot{1}, \dot{1}, \#, \dot{\#})$
f''_c	$(1, \#, \#, \dot{1}, \dot{1}, \#, \dot{\#})$
f''_d	$(\dot{1}, \#, \dot{\#}, 1, \#, \#, \#)$

Figure C.2: The F'_1 and F''_1 sets of the GKL rule.

3. **F'' may contain two-symbol schemata that render wildcard schemata with more wildcard symbols non-essential both in terms of describing the automaton and contribution to k_u.**

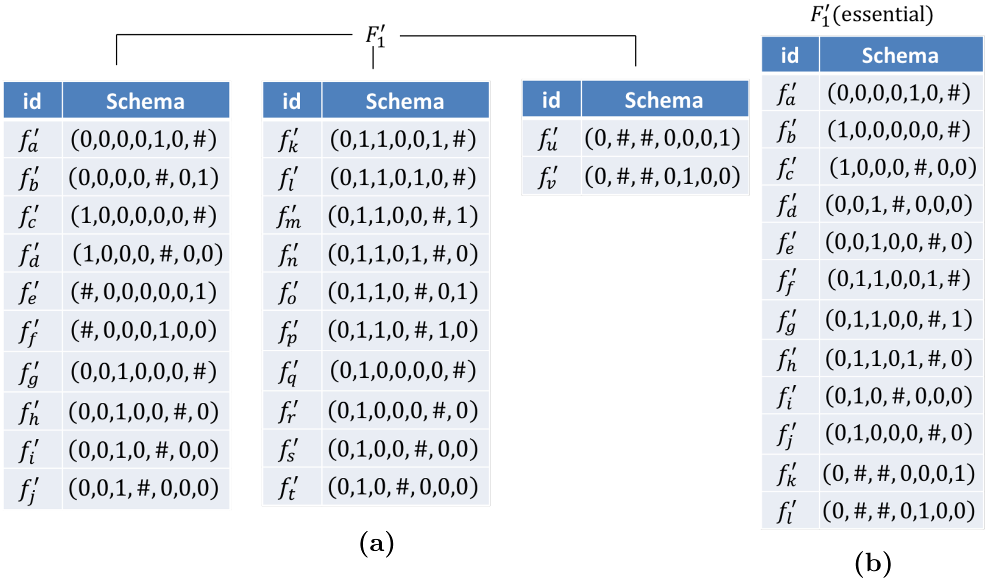
   In the previous example, the wildcard schemata with more wildcard symbols than the two-symbol schemata in F'' were essential for describing the automaton and also contributed to k_u. However, that need not always be the case. Here, we describe an example.

   Consider the automaton whose full and essential F'_1 sets are described in Fig. C.3. The essential version of F'_1, which is the smallest subset of the full version that redescribes all LUT entries F_1, contains f'_k and f'_l both of which have two wildcards. Being the essential wildcard schemata, they contribute to k_e. The corresponding full and essential versions of F_1'' are depicted in Fig. C.4. Notice here that the essential version of F_1'' does *not* contain the wildcard schemata with 2 wildcards, namely f_g'' and f_h'', that are present in the full F_1''. In fact, all two-symbol schemata in this set have exactly one wildcard. This is because every LUT entry redescribed by f_g'' or f_h'' is redescribed by some two-symbol schema with a larger dimension; all two-symbol schemata with permuting symbols in this set have a dimension of either $\log_2(5) \approx 2.32$ or $\log_2(6) \approx 2.58$, whereas the wildcard schemata have a dimension of 2. Thus, even though the wildcard schemata have more wildcard symbols than any of the other two-symbol schemata, they are rendered non-essential in F_1'' since they are neither needed to describe the automaton nor contribute to k_u.

However, the above is not a consequence of some artifact of the revised (§ 3.1) method of computing F'', at least in this case. That is, the current method of computing F'' as described in [88] also results in the rendering of the aforementioned wildcard schemata as non-essential[1]. The current full and essential versions of F_1'' are depicted in Fig. C.5. Note that the current full F_1'' is computed from the set of *all* wildcard schemata [88], that is the full F_1' (Fig. C.3(a)), not the essential F_1'. Even here, the two-symbol schema that contains two wildcard symbols, namely f_e'', present in the full F_1'' is not contained in the essential F_1''. This is because, every LUT entry redescribed by f_e'' is redescribed by some other two-symbol schema with more permutation symbols. Thus, f_e'' is non-essential because it is neither needed for describing the automaton nor does it contribute to k_s, that is, the k_s won't change even if we remove f_e'' from F_1''.

---

[1] The example automaton with $k = 7$ inputs described here is the simplest example we could generate that captures this phenomenon.



F'_1

id	Schema
f'_a	$(0,0,0,0,1,0,\#)$
f'_b	$(0,0,0,0,\#,0,1)$
f'_c	$(1,0,0,0,0,0,\#)$
f'_d	$(1,0,0,0,\#,0,0)$
f'_e	$(\#,0,0,0,0,0,1)$
f'_f	$(\#,0,0,0,1,0,0)$
f'_g	$(0,0,1,0,0,0,\#)$
f'_h	$(0,0,1,0,0,\#,0)$
f'_i	$(0,0,1,0,\#,0,0)$
f'_j	$(0,0,1,\#,0,0,0)$

id	Schema
f'_k	$(0,1,1,0,0,1,\#)$
f'_l	$(0,1,1,0,1,0,\#)$
f'_m	$(0,1,1,0,0,\#,1)$
f'_n	$(0,1,1,0,1,\#,0)$
f'_o	$(0,1,1,0,\#,0,1)$
f'_p	$(0,1,1,0,\#,1,0)$
f'_q	$(0,1,0,0,0,0,\#)$
f'_r	$(0,1,0,0,0,\#,0)$
f'_s	$(0,1,0,0,\#,0,0)$
f'_t	$(0,1,0,\#,0,0,0)$

id	Schema
f'_u	$(0,\#,\#,0,0,0,1)$
f'_v	$(0,\#,\#,0,1,0,0)$

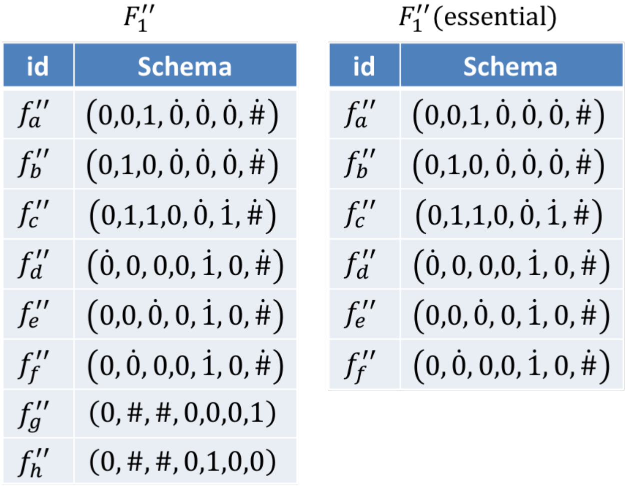
(a)

F'_1(essential)

id	Schema
f'_a	$(0,0,0,0,1,0,\#)$
f'_b	$(1,0,0,0,0,0,\#)$
f'_c	$(1,0,0,0,\#,0,0)$
f'_d	$(0,0,1,\#,0,0,0)$
f'_e	$(0,0,1,0,0,\#,0)$
f'_f	$(0,1,1,0,0,1,\#)$
f'_g	$(0,1,1,0,0,\#,1)$
f'_h	$(0,1,1,0,1,\#,0)$
f'_i	$(0,1,0,\#,0,0,0)$
f'_j	$(0,1,0,0,0,\#,0)$
f'_k	$(0,\#,\#,0,0,0,1)$
f'_l	$(0,\#,\#,0,1,0,0)$

(b)

Figure C.3: The full and the essential versions of F'_1 of an example automaton.



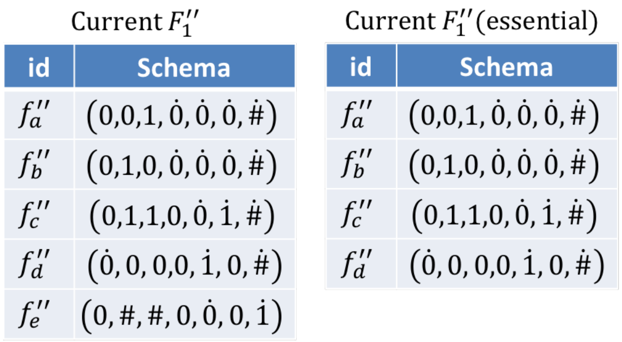
F''_1

id	Schema
f''_a	$(0,0,1,\dot{0},\dot{0},\dot{0},\dot{\#})$
f''_b	$(0,1,0,\dot{0},\dot{0},\dot{0},\dot{\#})$
f''_c	$(0,1,1,0,\dot{0},\dot{1},\dot{\#})$
f''_d	$(\dot{0},0,0,0,\dot{1},0,\dot{\#})$
f''_e	$(0,0,\dot{0},0,\dot{1},0,\dot{\#})$
f''_f	$(0,\dot{0},0,0,\dot{1},0,\dot{\#})$
f''_g	$(0,\#,\#,0,0,0,1)$
f''_h	$(0,\#,\#,0,1,0,0)$

F''_1(essential)

id	Schema
f''_a	$(0,0,1,\dot{0},\dot{0},\dot{0},\dot{\#})$
f''_b	$(0,1,0,\dot{0},\dot{0},\dot{0},\dot{\#})$
f''_c	$(0,1,1,0,\dot{0},\dot{1},\dot{\#})$
f''_d	$(\dot{0},0,0,0,\dot{1},0,\dot{\#})$
f''_e	$(0,0,\dot{0},0,\dot{1},0,\dot{\#})$
f''_f	$(0,\dot{0},0,0,\dot{1},0,\dot{\#})$

Figure C.4: The full and the essential versions of F''_1 obtained from the full F'_1 shown in Fig. C.3. These sets are used in the calculation of k_u (not shown).



Current F_1''

id	Schema
f_a''	$(0,0,1,\dot{0},\dot{0},\dot{0},\dot{\#})$
f_b''	$(0,1,0,\dot{0},\dot{0},\dot{0},\dot{\#})$
f_c''	$(0,1,1,0,\dot{0},\dot{1},\dot{\#})$
f_d''	$(\dot{0},0,0,0,\dot{1},0,\dot{\#})$
f_e''	$(0,\#,\#,0,\dot{0},0,\dot{1})$

Current F_1''(essential)

id	Schema
f_a''	$(0,0,1,\dot{0},\dot{0},\dot{0},\dot{\#})$
f_b''	$(0,1,0,\dot{0},\dot{0},\dot{0},\dot{\#})$
f_c''	$(0,1,1,0,\dot{0},\dot{1},\dot{\#})$
f_d''	$(\dot{0},0,0,0,\dot{1},0,\dot{\#})$

Figure C.5: The full and the essential versions of F_1'' obtained by the current method of computing two-symbol schemata as described in [88]. These sets are used in the calculation of k_s (not shown).

Curriculum Vitae

Santosh Manicka

santosh.manicka@gmail.com

Education

2009-2017: Ph.D. Informatics (Complex Systems track), Indiana University Bloomington, U.S.A.

2007-2008: M.Sc. Evolutionary and Adaptive Systems, University of Sussex, U.K.

1999-2003: B.E. Computer Science and Engineering, College of Engineering, Guindy, Anna University, India.

Theses

Ph.D. Thesis

Title: *The role of canalization in the spreading of perturbations in Boolean networks*

Advisor: Prof. Luis M. Rocha

Description: This dissertation demonstrates that the measures of partial canalization of Boolean functions, namely input redundancy and input symmetry, constitute more accurate control parameters for phase transitions in random Boolean networks than the mean in-degree. It further demonstrates the canalization levels of dynamically integrated Boolean networks, that describe the logic at an arbitrary timescale of the dynamics, are even more accurate predictors of the long-term dynamical regime.

M.Sc. Thesis

Title: *Towards Ultrastable Robotics: An initial exploration using Programmable Springs*

Advisors: Profs. Ezequiel Di Paolo and Inman Harvey

Description: This dissertation investigates the concept of 'ultrastability', first formulated by Ross Ashby, as a potential homeostatic adaptation mechanism in a real robotic actuator driven by a non-linear programmable 'force profile'. The goal of the actuator is to coax a metal ball to stay in the center of a see-saw bridge, by dynamically generating a suitable force profile using the method of ultrastability which fine-tunes the parameters of a dynamical system based on the desired properties of its phase space.

Research Interests

- Dynamics of Boolean networks
- Canalization and symmetry in Boolean functions
- Agent-based modeling for Systems biology
- Relationship between discrete and continuous dynamical systems

Courses

Ph.D.: Dynamical systems, Information theory, Networks of the brain, Math and logic, Statistical computing, Statistical learning, Philosophical foundations of cognitive and information sciences, and University science teaching.

M.Sc.: Artificial life, Adaptive systems, Simulation of adaptive behavior, Computational neuroscience and Neural networks.

Conferences attended

2016: 'Quantitative biology' summer school, Colorado state university

2012: International conference on Artificial life, Michigan state university, U.S.A. Presented the paper "Analysis of evolved agents performing referential communication"

2010: 'Dynamic field theory' summer school, University of Iowa

2008: International conference on Artificial life, University of Southampton, U.K. Presented the paper "'Psychoanalysis' of a minimal agent"

Achievements and awards

- Awarded full scholarship to attend the 'Quantitative biology' summer school, Colorado state university, 2016
- Awarded the IGERT affiliate summer research grants, Indiana University Bloomington, 2012-2015
- Awarded the School of Informatics PhD fellowship, Indiana University Bloomington, 2009-2011

- Awarded travel assistantship for summer research, Complex adaptive systems and computational biology lab, Gulbenkian institute of science, Portugal, 2010-2011
- Awarded full scholarship to attend the ‘Dynamic field theory’ summer school, University of Iowa, 2010
- Winner of ‘On the spot’ award three consecutive times for excellent analytic skills, Infosys Technologies Ltd., 2003-2006

Experience

Teaching

2011-2016: Associate Instructor, Indiana University Bloomington, U.S.A. Ran labs and gave lectures for various undergraduate-level courses including Network science, Math foundations of Informatics, Bio-inspired computing, Performance analytics and Information representation.

Vocational

2008-2009: Senior Software Engineer, Mindlab International, Brighton, U.K. Analyzed EEG recordings using Matlab software for neuromarketing purposes.

2003-2006: Programmer Analyst, Infosys Technologies, Chennai and Kansas city, India and U.S.A. Developed and maintained mainframe software for various clients in the retail domain.

Skills

Programming

- R, Python, Matlab, Mathematica, C++

Publications

- Santosh Manicka. Analysis of evolved agents performing referential communication. In *Artificial Life*, volume 13, pages 393–400, 2012
- Santosh Manicka and Ezequiel A Di Paolo. Local ultrastability in a real system based on programmable springs. In *Advances in Artificial Life. Darwin Meets von Neumann*, pages 91–98. Springer, 2009
- Santosh Manicka and Inman Harvey. 'Psychoanalysis' of a minimal agent. In *ALIFE*, pages 382–389, 2008
- D Manjula, Santhosh Kulandaiyan, **Sudarshan, Santosh**, Ashok Francis, and TV Geetha. Semantics based information retrieval using conceptual indexing of documents. In *Intelligent Data Engineering and Automated Learning*, pages 685–692. Springer, 2003
- M. Marques-Pita, S. Manicka, and Rocha L.M. Effective connectivity as a control parameter for random boolean networks. *In preparation for PNAS*, 2016.

References

1. Luis M. Rocha

   Professor

   Complex Systems Program

School of Informatics and Computing

Indiana University Bloomington

rocha@indiana.edu

+1 (812) 856 1832

2. Randall D. Beer

   Provost Professor

   Cognitive Science program

   School of Informatics and Computing

   Indiana University Bloomington

   rdbeer@indiana.edu

   +1 (812) 856 0873

3. Olaf Sporns

   Distinguished Professor

   Provost Professor

   Department of Psychological and Brain Sciences

   Indiana University Bloomington

   osporns@indiana.edu

   +1 (812) 855 2772